



1
2
3
4

Document Number: DSP1119

Date: 2013-05-20

Version: 1.0.0b

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: 2013-10-31

Target version for DMTF Standard: 2.34.0

6 **Document Type: Specification**

7 **Document Status: Work in Progress**

8 **Document Language: en-US**

9 Copyright Notice

10 Copyright © 2013 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25 withdrawn or modified after publication, and shall be indemnified and held harmless by any party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	Foreword	7
33	Introduction.....	8
34	Document conventions.....	8
35	1 Scope	9
36	2 Normative references	9
37	3 Terms and definitions	10
38	4 Symbols and abbreviated terms.....	10
39	5 Synopsis	11
40	6 Description	12
41	7 Implementation.....	14
42	7.1 CIM_AffectedJobElement	14
43	7.1.1 CIM_AffectedJobElement.ElementEffects.....	14
44	7.2 CIM_ConcreteJob	14
45	7.2.1 CIM_ConcreteJob.InstanceID.....	14
46	7.2.2 CIM_ConcreteJob.Name	15
47	7.2.3 CIM_ConcreteJob.JobState.....	15
48	7.2.4 CIM_ConcreteJob.DeleteOnCompletion	15
49	7.2.5 CIM_ConcreteJob.TimeBeforeRemoval.....	15
50	7.2.6 CIM_ConcreteJob.StartTime	15
51	7.2.7 CIM_ConcreteJob.ElapsedTime.....	16
52	7.2.8 CIM_ConcreteJob.PercentComplete	16
53	7.2.9 CIM_ConcreteJob.TimeOfLastStateChange	16
54	7.3 CIM_DiagnosticServiceJobCapabilities	16
55	7.3.1 CIM_DiagnosticServiceJobCapabilities.InstanceID.....	17
56	7.3.2 CIM_DiagnosticServiceJobCapabilities.ElementName	17
57	7.3.3 CIM_DiagnosticServiceJobCapabilities.DeleteJobSupported	17
58	7.3.4 CIM_DiagnosticServiceJobCapabilities.RequestedStatesSupported	17
59	7.3.5 CIM_DiagnosticServiceJobCapabilities.InteractiveTimeoutMax	17
60	7.3.6 CIM_DiagnosticServiceJobCapabilities.DefaultValuesSupported.....	18
61	7.3.7 CIM_DiagnosticServiceJobCapabilities.ClientRetriesMax.....	18
62	7.3.8 CIM_DiagnosticServiceJobCapabilities.CleanupInterval.....	18
63	7.3.9 CIM_DiagnosticServiceJobCapabilities.SilentModeSupported	18
64	7.4 CIM_JobSettingData (Client)	18
65	7.4.1 CIM_JobSettingData.InstanceID	19
66	7.4.2 CIM_JobSettingData.DeleteOnCompletion	19
67	7.4.3 CIM_JobSettingData.InteractiveTimeout.....	19
68	7.4.4 CIM_JobSettingData.TerminateOnTimeout	19
69	7.4.5 CIM_JobSettingData.DefaultInputValues	20
70	7.4.6 CIM_JobSettingData.DefaultInputNames.....	20
71	7.4.7 CIM_JobSettingData.ClientRetries	20
72	7.4.8 CIM_JobSettingData.RunInSilentMode	20
73	7.5 CIM_JobSettingData (Default)	20
74	7.5.1 CIM_JobSettingData.InstanceID	21
75	7.5.2 CIM_JobSettingData.DeleteOnCompletion	21
76	7.5.3 CIM_JobSettingData.InteractiveTimeout.....	21
77	7.5.4 CIM_JobSettingData.TerminateOnTimeout	22
78	7.5.5 CIM_JobSettingData.DefaultInputValues	22
79	7.5.6 CIM_JobSettingData.DefaultInputNames.....	22
80	7.5.7 CIM_JobSettingData.ClientRetries	22
81	7.5.8 CIM_JobSettingData.RunInSilentMode	22
82	7.6 Interactive options.....	23

83	7.7	Job deletion options	25
84	7.8	Diagnostic Job Control Profile indications support	28
85	7.8.1	CIM_IndicationFilter (StaticIndicationFilter)	28
86	7.8.2	CIM_FilterCollection (ProfileSpecificFilterCollection)	28
87	7.8.3	CIM_MemberOfCollection (IndicationFilterInFilterCollection)	28
88	7.8.4	CIM_OwningCollectionElement (IndicationServiceOfFilterCollection)	29
89	7.9	Diagnostics job control alert indications and standard messages	29
90	7.9.1	DIAG9 – Test continued after last interactive timeout using default values	29
91	7.9.2	DIAG12 – Job could not be started.....	29
92	7.9.3	DIAG19 – Test killed by client.....	30
93	7.9.4	DIAG20 – Test terminated by client.....	30
94	7.9.5	DIAG21 – Test suspended by client	31
95	7.9.6	DIAG34 – Request for inputs	32
96	7.9.7	DIAG35 – Request for action.....	32
97	7.9.8	DIAG36 – Test killed by test	33
98	7.9.9	DIAG37 – Test terminated by test	33
99	7.9.10	DIAG38 – Test resumed by client.....	34
100	7.9.11	DIAG39 – JobSettings reset	34
101	7.9.12	DIAG40 – JobSettings defaults not used.....	35
102	7.9.13	DIAG48 – Test continued after an interim interactive timeout	36
103	7.9.14	DIAG49 – Test terminated after an interactive timeout	36
104	8	Methods.....	37
105	8.1	Profile conventions for operations	37
106	8.2	CIM_ConcreteJob	37
107	8.2.1	CIM_ConcreteJob.RequestStateChange()	37
108	8.2.2	CIM_ConcreteJob.ResumeWithInput()	38
109	8.2.3	CIM_ConcreteJob.ResumeWithAction()	39
110	8.3	CIM_DiagnosticServiceJobCapabilities	40
111	8.3.1	CreateGoalSettings()	40
112	8.4	CIM_MethodResult	41
113	8.5	CIM_OwningJobElement	41
114	8.6	CIM_AffectedJobElement	41
115	8.7	CIM_AssociatedJobMethodResult.....	41
116	8.8	CIM_HostedDependency.....	41
117	8.9	CIM_RegisteredProfile.....	41
118	8.10	CIM_JobSettingData	41
119	8.11	CIM_ElementSettingData	42
120	8.12	CIM_ElementCapabilities	42
121	8.13	CIM_DiagnosticTest.RunDiagnosticService()	42
122	9	Use cases (informative).....	43
123	9.1	Use case summary	43
124	9.2	User input required	45
125	9.2.1	Single prompt and response has a valid value	45
126	9.2.2	Single prompt and response has multiple valid values.....	46
127	9.2.3	Multiple prompts and responses required with partial test execution after each.....	46
128	9.2.4	Client does not respond to a prompt.....	47
129	9.2.5	Client responds with an invalid value.....	47
130	9.2.6	Client does not respond with enough valid values	47
131	9.3	User action required.....	48
132	9.3.1	Single prompt and response required.....	48
133	9.3.2	Multiple prompts and responses required before running the test	48
134	9.3.3	Multiple prompts and responses required with partial test execution after each.....	49
135	9.3.4	Client does not respond to a prompt.....	49
136	9.4	Silent mode operation	50
137	9.4.1	Running an Interactive Test in Silent Mode.....	50
138	9.4.2	Running Silent Mode with invalid default values	50

139 9.5 Finding diagnostic jobs 51
 140 9.5.1 Finding all diagnostic tests executed on a system 51
 141 9.5.2 Finding all diagnostic tests executed against a ManagedElement..... 51
 142 9.6 Configuring a diagnostic job 52
 143 9.6.1 Getting the default job settings 52
 144 9.6.2 Creating the job settings 52
 145 9.7 Execute and control a job for a diagnostic test 52
 146 9.7.1 Suspend a job for a diagnostic test..... 53
 147 9.7.2 Resume a job for a diagnostic test 53
 148 9.7.3 Terminate a job for a diagnostic test..... 54
 149 9.7.4 Kill a job for a diagnostic test 54
 150 9.8 Delete a job for a diagnostic test 55
 151 9.8.1 Client deletes a job for a diagnostic test..... 55
 152 9.8.2 Provider deletes a job 55
 153 10 CIM elements 56
 154 10.1 CIM_AffectedJobElement 58
 155 10.2 CIM_ConcreteJob 58
 156 10.3 CIM_DiagnosticServiceJobCapabilities 59
 157 10.4 CIM_ElementCapabilities (Job) 59
 158 10.5 CIM_ElementSettingData (Default JobSettingData)..... 59
 159 10.6 CIM_FilterCollection (ProfileSpecificFilterCollection) 60
 160 10.7 CIM_HostedDependency..... 60
 161 10.8 CIM_IndicationFilter (StaticIndicationFilter) 60
 162 10.9 CIM_JobSettingData (Client) 61
 163 10.10 CIM_JobSettingData (Default) 61
 164 10.11 CIM_MemberOfCollection (ProfileSpecificMemberOfCollection) 62
 165 10.12 CIM_OwningCollectionElement 62
 166 10.13 CIM_OwningJobElement 62
 167 10.14 CIM_RegisteredProfile 63
 168 ANNEX A (informative) Change log 64
 169

170 **Figures**

171	Figure 1 – Diagnostic Job Control Profile: Profile class diagram	13
-----	--	----

172

173 **Tables**

174	Table 1 – Referenced profiles	12
175	Table 2 – OperationalStatus to JobState mapping	15
176	Table 3 – Interactive options	24
177	Table 4 – Job deletion options	27
178	Table 5 – ResumeWithInput() method: Return code values	39
179	Table 6 – ResumeWithInput() method: Parameters	39
180	Table 7 – ResumeWithAction() method: Return code values	40
181	Table 8 – CreateGoalSettings() method: Return code values	40
182	Table 9 – CreateGoalSettings() method: Parameters	41
183	Table 10 – Operations: CIM_JobSettingData	42
184	Table 11 – Operations: CIM_ElementSettingData	42
185	Table 12 – Operations: CIM_ElementCapabilities	42
186	Table 13 – Job settings options	43
187	Table 14 – Diagnostic test use cases	44
188	Table 15 – CIM Elements: Diagnostic Job Control Profile	56
189	Table 16 – Class: CIM_AffectedJobElement	58
190	Table 17 – Class: CIM_ConcreteJob	58
191	Table 18 – Class: CIM_DiagnosticServiceJobCapabilities	59
192	Table 19 – Class: CIM_ElementCapabilities	59
193	Table 20 – Class: CIM_ElementSettingData	60
194	Table 21 – Class: CIM_FilterCollection	60
195	Table 22 – Class: CIM_HostedDependency	60
196	Table 23 – Class: CIM_IndicationFilter (StaticIndicationFilter)	61
197	Table 24 – Class: CIM_JobSettingData (Client)	61
198	Table 25 – Class: CIM_JobSettingData (Default)	61
199	Table 26 – Class: CIM_MemberOfCollection	62
200	Table 27 – Class: CIM_OwningCollectionElement	62
201	Table 28 – Class: CIM_OwningJobElement	63
202	Table 29 – Class: CIM_RegisteredProfile	63
203		

204

Foreword

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

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

209 **Acknowledgments**

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

- 211 • Dave Barrett – Emulex Corporation
- 212 • Rodney Brown – IBM Corporation
- 213 • Carl Chan – WBEM Solutions, Inc.
- 214 • Peter Lamanna – EMC Corporation
- 215 • Mike Walker – Storage Networking Industry Association

216

Introduction

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

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

225 Document conventions

226 Typographical conventions

227 The following typographical conventions are used in this document:

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

230 ABNF usage conventions

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

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

235 Experimental material

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

241 The following typographical convention indicates experimental material:

242 **EXPERIMENTAL**

243 Experimental material appears here.

244 **EXPERIMENTAL**

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

247

Diagnostic Job Control Profile

248 1 Scope

249 The *Diagnostic Job Control Profile* is a specialization of the *Job Control Profile* ([DSP1103](#)) that extends
250 the profile by defining the job control functions used to monitor and interact with diagnostic tests.

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

253 2 Normative references

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

258 DMTF DSP0004, *CIM Infrastructure Specification 2.6*,
259 http://dmtof.org/sites/default/files/standards/documents/DSP0004_2.6.pdf

260 DMTF DSP0200, *CIM Operations over HTTP 1.3*,
261 http://dmtof.org/sites/default/files/standards/documents/DSP0200_1.3.pdf

262 DMTF DSP1001, *Management Profile Specification Usage Guide 1.0*,
263 http://dmtof.org/sites/default/files/standards/documents/DSP1001_1.0.pdf

264 DMTF DSP1002, *Diagnostics Profile Specification 2.0*,
265 http://dmtof.org/sites/default/files/standards/documents/DSP1002_2.0.pdf

266 DMTF DSP1033, *Profile Registration Profile 1.0*,
267 http://dmtof.org/sites/default/files/standards/documents/DSP1033_1.0.pdf

268 DMTF DSP1054, *Indications Profile 1.2*,
269 http://dmtof.org/sites/default/files/standards/documents/DSP1054_1.2.pdf

270 DMTF DSP1103, *Job Control Profile 1.0.0*,
271 http://dmtof.org/sites/default/files/standards/documents/DSP1103_1.0.pdf

272 DMTF DSP1104, *Fibre Channel Host Bus Adapter Diagnostics Profile 1.0*
273 http://dmtof.org/sites/default/files/standards/documents/DSP1104_1.0.0.pdf

274 DMTF DSP8055, *Diagnostics Message Registry 1.0.0a*,
275 http://dmtof.org/sites/default/files/standards/documents/DSP8055_1.0a.xml

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

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

280 3 Terms and definitions

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

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

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

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

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

295 3.1

296 job

297 a task or thread or execution under which an operation will be run.

298 3.2

299 organization

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

301 3.3

302 test

303 a test is task that performs parameter and environment checking before launching a job to execute
304 diagnostic operations against a target element.

305 4 Symbols and abbreviated terms

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

307 4.1

308 CDM

309 Common Diagnostic Model

310 4.2

311 CIM

312 Common Information Model

313 4.3

314 CIMOM

315 CIM Object Manager

316 4.4

317 CQL

318 CIM Query Language

319 **4.5**
 320 **ME**
 321 Managed Element
 322 **4.6**
 323 **MOF**
 324 Managed Object Format
 325 **4.7**
 326 **OS**
 327 Operating System
 328 **4.8**
 329 **QoS**
 330 Quality of Service
 331 **4.9**
 332 **URI**
 333 Uniform Resource Identifier
 334 **4.10**
 335 **WBEM**
 336 Web-Based Enterprise Management

337 **5 Synopsis**

338 **Profile name:** Diagnostics Job Control

339 **Version:** 1.0.0b

340 **Organization:** DMTF

341 **CIM schema version:** 2.34

342 **Central class:** CIM_ConcreteJob

343 **Scoping class:** CIM_System

344 **Specializes:** Job Control Profile 1.0.0

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

348 The Central Instance of this profile shall be an instance of CIM_ConcreteJob. The Scoping Instance shall
 349 be the instance of CIM_System (the central instance of the referencing profile) with which the Central
 350 Instance (the instance of CIM_ConcreteJob) is associated through CIM_HostedDependency. The
 351 CIM_System is the system running the CIM_DiagnosticTest and its associated CIM_ConcreteJob.

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

353

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

354 6 Description

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

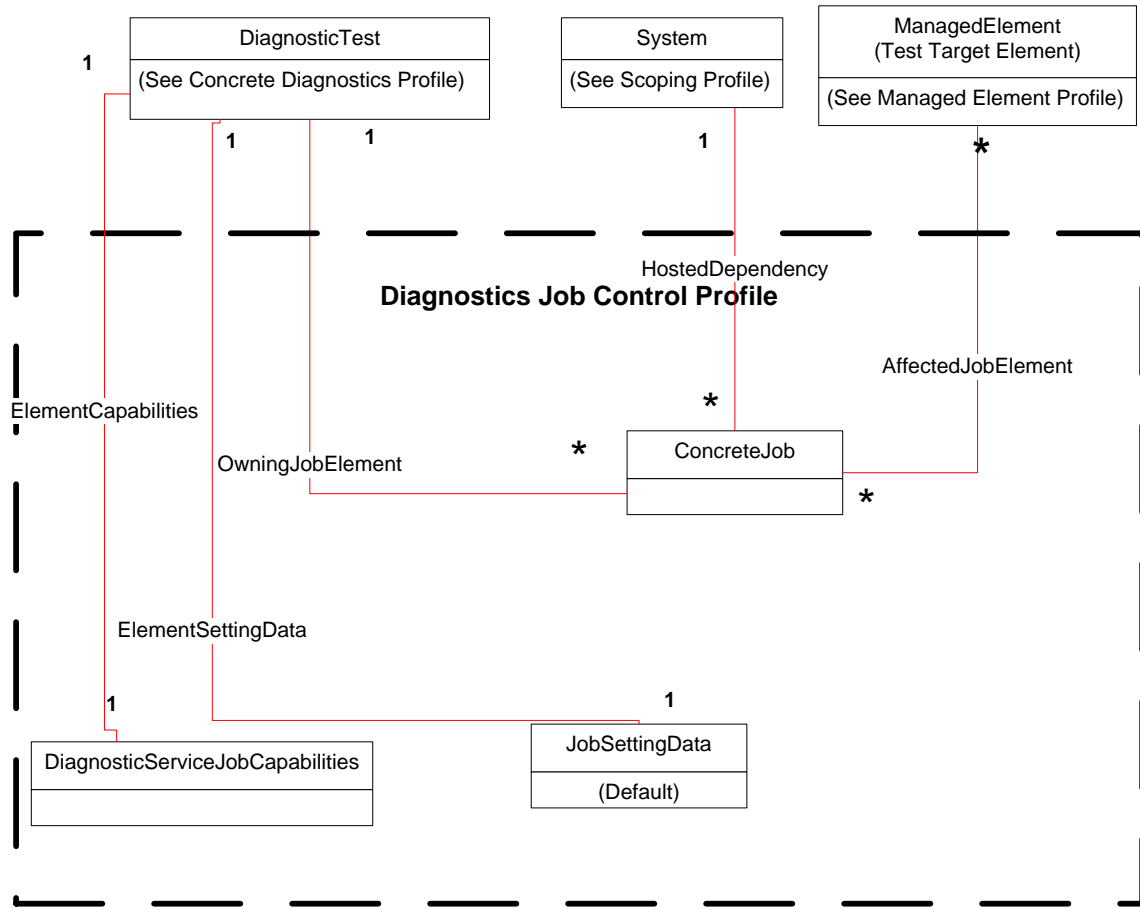
360 Specifically, a client application invokes `CIM_DiagnosticTest.RunDiagnosticService()` to start a diagnostic
 361 test. A user may optionally pass an embedded instance of `CIM_JobSettingData` as an input parameter to
 362 specify the behavior of the associated `CIM_ConcreteJob` instance. A `CIM_ConcreteJob` instance is
 363 created when a diagnostic test starts. When the `CIM_ConcreteJob` instance is deleted it is controlled by
 364 the values of the properties in `CIM_ConcreteJob` and `CIM_JobSettingData`.

365 To start a diagnostic test, the client application calls `CIM_DiagnosticTest.RunDiagnosticService()` which
 366 returns 0 (Success) and the object path of a single `CIM_ConcreteJob` instance. The `CIM_ConcreteJob`
 367 manages the diagnostic test execution. Additionally, a client can monitor and interact with the diagnostic
 368 test execution or workflow via the returned `CIM_ConcreteJob` instance.

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

371 After the diagnostic test is completed, its `CIM_ConcreteJob` instance will persist for a predetermined
 372 length of time before deletion. The same diagnostic test could start again, creating another
 373 `CIM_ConcreteJob` instance before the previous `CIM_ConcreteJob` instance is deleted. For this reason,
 374 the cardinality of `CIM_OwningJobElement` is one-to-many.

375



376

377

Figure 1 – Diagnostic Job Control Profile: Profile class diagram

378 The Referencing Profile for DiagnosticTest is a diagnostic component profile. For example, if the
 379 Referencing Profile is the *Fibre Channel Host Bus Adapter Diagnostics Profile* ([DSP1104](#)), the
 380 DiagnosticTest is a subclass called FCHBADiagnosticTest. The Referencing Profile for the
 381 ManagedElement is a profile that contains the ManagedElement and that profile should reference the
 382 profile with the DiagnosticTest. For the [DSP1104](#), the ManagedElement is CIM_PortController in a
 383 system profile. That system profile would reference the Fibre Channel Host Bus Adapter Diagnostic
 384 Profile, which in turn would reference the Diagnostic Job Control Profile. The System in Figure 1 would be
 385 a system in a profile that references the ManagedElement profile.

386

387 7 Implementation

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

390 7.1 CIM_AffectedJobElement

391 This subclause defines the properties of the CIM_AffectedJobElement class, which associates a
392 CIM_ConcreteJob instance to the CIM_ManagedElement instances that are affected by the job.
393 Minimally, this shall contain the ManagedElement under test. However, it might also include other
394 managed elements that could be affected. For example, an FC HBA self-test would affect the FC HBA,
395 but might also affect the FC ports on the HBA. In addition, it might impact the system in which the FC
396 HBA is a component.

397 7.1.1 CIM_AffectedJobElement.ElementEffects

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

401 7.2 CIM_ConcreteJob

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

407 7.2.1 CIM_ConcreteJob.InstanceID

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

410 <OrgID>:<LocalID>

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

414 The purpose for <LocalID> is to provide some form of uniqueness within the context of separate test
415 instances over an extended period of time. This uniqueness of the test execution identifier could include
416 multiple executions of the same test or multiple executions of multiple tests. In practice, <LocalID> could
417 be an incremented counter or a timestamp in combination with other test names and job identifiers or
418 monikers.

419 A unique <LocalID> allows a user to easily retrieve test results from the diagnostic log for a specific test
420 execution because the InstanceID values of CIM_ConcreteJob, and the subclasses of
421 CIM_DiagnosticRecord are closely related. In [DSP1002](#), see Figure 5 and use case 9.8.4
422 GetDiagnosticExecutionFinalResults for further information.

423 Specifically, CIM_DiagnosticRecord.InstanceID has the same value as its related
424 CIM_ConcreteJob.InstanceID with an appended record number identifier. As an example, if
425 CIM_ConcreteJob.InstanceID has the form "Widget:<StartTime>", for the third record,
426 CIM_DiagnosticRecord.InstanceID has the form "Widget:<StartTime>:3", where <StartTime> is the value
427 of CIM_ConcreteJob.StartTime.

428 **7.2.2 CIM_ConcreteJob.Name**

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

431 **7.2.3 CIM_ConcreteJob.JobState**

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

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

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

447 **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

448 **7.2.4 CIM_ConcreteJob.DeleteOnCompletion**

449 This profile extends [DSP1103](#) to define that the default value shall be TRUE. This boolean property
 450 indicates whether the CIM_ConcreteJob instance associated to a diagnostic test execution is
 451 automatically deleted when test execution is completed after a configurable time period. See subclause
 452 7.2.5 for further implementation details.

453 **7.2.5 CIM_ConcreteJob.TimeBeforeRemoval**

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

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

457 [DSP1103](#) recommends a value of five or more minutes. For diagnostic test environments, clients should
 458 use a value that balances the time required to collect the results against the load on the system.. For
 459 example, in lightly loaded environments it should be possible to choose a value close to test completion
 460 time whereas in heavily loaded environments it would be safer to configure a value a multiple of that time
 461 period.

462 **7.2.6 CIM_ConcreteJob.StartTime**

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

466 7.2.7 CIM_ConcreteJob.ElapsedTime

467 For this profile, the value of this interval datetime property shall be updated at a vendor-defined interval. A
468 client can monitor this property at a client-defined interval. When the property changes its value, the client
469 knows that the test is still making progress. It is recommended that it be synchronized with the updating of
470 PercentComplete.

471 7.2.8 CIM_ConcreteJob.PercentComplete

472 In addition to the requirements specified in [DSP1103](#) and CIM_ConcreteJob.ElapsedTime, this profile
473 uses PercentComplete to show the amount of testing done in terms of actual percent complete. Service
474 implementations should update this property within a reasonable time of becoming aware of a progress
475 change. It is recommended that it be synchronized with the updating of ElapsedTime. If progress cannot
476 be determined with that reasonable amount of time it should be set to 50 percent. It shall be set to 100
477 percent only when the test is complete. It shall not be set to 100 percent if the test stops for any other
478 reason (for example, the test stopped or was killed by user, the test exited due to a critical failure, or the
479 test found an error and HaltOnError is TRUE) because the actual percent complete is not 100 percent.

480 7.2.9 CIM_ConcreteJob.TimeOfLastStateChange

481 The date and time when the state of the Job last changed (via RequestedStateChange). If the state of the
482 Job has not changed and this property is populated, then it shall be set to a 0 interval value. If a state
483 change was requested, but rejected or not yet processed, the property shall not be updated. If a state
484 change was requested, accepted and processed successfully, then the value shall be the date and time
485 of the successful completion of the state change.

486 EXPERIMENTAL

487 7.3 CIM_DiagnosticServiceJobCapabilities

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

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

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

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

508 **7.3.1 CIM_DiagnosticServiceJobCapabilities.InstanceID**

509 CIM_DiagnosticServiceJobCapabilities.InstanceID should be constructed using the following preferred
510 algorithm:

511 <OrgID>:<LocalID>

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

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

517 **7.3.2 CIM_DiagnosticServiceJobCapabilities.ElementName**

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

520 **7.3.3 CIM_DiagnosticServiceJobCapabilities.DeleteJobSupported**

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

525 If the value of DeleteJobSupported is FALSE, a client cannot perform a DeleteInstance operation on a
526 CIM_ConcreteJob instance associated to its CIM_DiagnosticTest instance. The default
527 CIM_JobSettingData.DeleteOnCompletion property shall have the value TRUE. In addition, a client shall
528 not set the value of the DeleteOnCompletion property in the CIM_JobSettingData instance that it passes
529 as the JobSettings parameter to the CIM_DiagnosticTest.RunDiagnosticService() extrinsic method.

530 If the value of DeleteJobSupported is TRUE, a client can perform a DeleteInstance operation on a
531 CIM_ConcreteJob instance associated to its CIM_DiagnosticTest instance. The default
532 CIM_JobSettingData.DeleteOnCompletion property may be TRUE or FALSE. In addition, a client may set
533 the value of the DeleteOnCompletion property in the CIM_JobSettingData instance that it passes as the
534 JobSettings parameter to the CIM_DiagnosticTest.RunDiagnosticService() extrinsic method.

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

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

540 **7.3.4 CIM_DiagnosticServiceJobCapabilities.RequestedStatesSupported**

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

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

546 **7.3.5 CIM_DiagnosticServiceJobCapabilities.InteractiveTimeoutMax**

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

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

552 If a client passes an instance of CIM_JobSettingData as the JobSettings parameter to the
553 CIM_DiagnosticTest.RunDiagnosticService() extrinsic method, and the value of
554 CIM_JobSettingData.InteractiveTimeout exceeds the value of
555 CIM_DiagnosticServiceJobCapabilities.InteractiveTimeoutMax, the associated CIM_ConcreteJob
556 instance shall use the value of CIM_DiagnosticServiceJobCapabilities.InteractiveTimeoutMax

557 **7.3.6 CIM_DiagnosticsServiceJobCapabilities.DefaultValuesSupported**

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

562 When the value is TRUE, the values of CIM_JobSettingData.DefaultInputValues and
563 JobSettingData.DefaultInputNames are used as needed by the CIM_ConcreteJob instance. Otherwise,
564 the value shall have the value FALSE.

565 **7.3.7 CIM_DiagnosticsServiceJobCapabilities.ClientRetriesMax**

566 This property shall have a value if the CIM_DiagnosticTest.Characteristics property contains the value of
567 3 (Is Interactive). Otherwise, it is ignored.

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

571 **7.3.8 CIM_DiagnosticsServiceJobCapabilities.CleanupInterval**

572 As described in [DSP1103](#), if the value of the CIM_ConcreteJob.DeleteOnCompletion property is FALSE,
573 the job associated to the diagnostic test execution shall remain until it is explicitly deleted. When the value
574 of the CIM_ConcreteJob.DeleteOnCompletion property is FALSE, the job, after completion, will remain
575 until it is explicitly deleted by the client. The CleanupInterval datetime property insures that the job will be
576 deleted should the client fail to do so. It defines the time interval before the job is removed.

577 **7.3.9 CIM_DiagnosticServiceJobCapabilities.SilentModeSupported**

578 If the value of the property is TRUE, the interactive diagnostic test is capable of running without prompting
579 the client for responses. Instead, the test uses the default input argument values defined in
580 CIM_JobSettingData. If the value of the property is FALSE, the interactive diagnostic test shall prompt the
581 client for responses.

582 **EXPERIMENTAL**

583 **7.4 CIM_JobSettingData (Client)**

584 This subclause indicates the properties of the CIM_JobSettingData class that may be used by a client as
585 the JobSettings parameter when invoking the CIM_DiagnosticTest.RunDiagnosticService() extrinsic
586 method to start a diagnostic test. An instance of this class controls the execution of CIM_ConcreteJob
587 instance related to the executing diagnostic test.

588 If CIM_DiagnosticServiceJobCapabilities is implemented, a client can create a CIM_JobSettingData
589 (Client) instance containing the default values specified in the CIM_DiagnosticServiceJobCapabilities

590 instance by invoking the CIM_DiagnosticServiceJobCapabilities.CreateJobSettings() extrinsic method.
591 See subclause 8.3.1 for further information.

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

597 For interactive diagnostic tests that will wait for a client response, the CIM_DiagnosticTest.Characteristics
598 property shall contain the value 3 (Is Interactive). An interactive test may define the time interval it shall
599 wait for a client to respond. While waiting for the client to respond, the CIM_ConcreteJob.JobState
600 property associated to the diagnostic test has the value of 12 (Query Pending). If the client fails to
601 respond within the specified time interval, the diagnostic test may terminate, resume using default
602 responses, or wait another time interval. The CIM_JobSettingData properties InteractiveTimeout,
603 TerminateOnTimeout, DefaultInputValues, and DefaultInputNames control such behavior. Thus, when the
604 CIM_DiagnosticTest.Characteristics property contains the value 3 (Is Interactive) and the value of
605 CIM_DiagnosticServiceJobCapabilities.DefaultValuesSupported is TRUE, these properties shall have a
606 value. Otherwise, they are ignored.

607 **7.4.1 CIM_JobSettingData.InstanceID**

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

609 <OrgID>:<LocalID>

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

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

615 **7.4.2 CIM_JobSettingData.DeleteOnCompletion**

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

620 NOTE When the value of the CIM_DiagnosticServiceJobCapabilities.DeleteJobSupported property is FALSE, the
621 value of CIM_JobSettingData.DeleteOnCompletion shall have the value TRUE.

622 **EXPERIMENTAL**

623 **7.4.3 CIM_JobSettingData.InteractiveTimeout**

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

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

628 **7.4.4 CIM_JobSettingData.TerminateOnTimeout**

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

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

635 **7.4.5 CIM_JobSettingData.DefaultInputValues**

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

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

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

644 **7.4.6 CIM_JobSettingData.DefaultInputNames**

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

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

652 **7.4.7 CIM_JobSettingData.ClientRetries**

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

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

660 **7.4.8 CIM_JobSettingData.RunInSilentMode**

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

666 If CIM_DiagnosticServiceJobCapabilities.SilentModeSupported has the value of FALSE, this property is
667 ignored.

668 **EXPERIMENTAL**

669 **7.5 CIM_JobSettingData (Default)**

670 This subclause specifies the properties of the default CIM_JobSettingData class. An instance of this class
671 controls the execution of a diagnostic test job. This class is optional. If it is implemented, a single instance

672 shall represent the CIM_JobSettingData (Default) for the CIM_DiagnosticTest. This instance is identified
 673 by CIM_ElementSettingData with IsDefault="true" between the instance and the CIM_DiagnosticTest
 674 instance. A different default CIM_JobSettingData instance may be defined for each CIM_DiagnosticTest
 675 test type. For example, each of the different CPU diagnostic tests may define a different set of default
 676 CIM_JobSettingData values.

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

683 If CIM_DiagnosticServiceJobCapabilities is implemented, the client should refer to the
 684 CIM_DiagnosticServiceJobCapabilities instance to see what restrictions the implementation may impose
 685 on the client providing the CIM_JobSettingData (Client) embedded instance.

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

690 **7.5.1 CIM_JobSettingData.InstanceID**

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

692 <OrgID>:<LocalID>

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

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

698 **7.5.2 CIM_JobSettingData.DeleteOnCompletion**

699 This boolean property indicates whether the job should be automatically deleted upon completion. The
 700 property is mandatory. When the value is TRUE, the job shall be deleted after the
 701 CIM_ConcreteJob.TimeBeforeRemoval time interval. When the value is FALSE, the job shall be deleted
 702 by a DeleteInstance operation.

703 NOTE When the value of the CIM_DiagnosticServiceJobCapabilities.DeleteJobSupported property is FALSE, the
 704 value of CIM_JobSettingData.DeleteOnCompletion shall have the value TRUE and the jobs created will always be
 705 removed after the TimeBeforeRemoval time interval.

706 **EXPERIMENTAL**

707 **7.5.3 CIM_JobSettingData.InteractiveTimeout**

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

710 NOTE If JobSettings parameter of the RunDiagnosticService is NULL (not supplied), then the default
 711 JobSettingData value for InteractiveTimeout shall be used.

712 When an interactive diagnostic test prompts a client for a response, the test may define the time interval it
 713 shall wait for a client to respond. While waiting for the client to respond, the CIM_ConcreteJob.JobState

714 property associated to the diagnostic test shall have a value of 12 (Query Pending). This property
715 indicates the time interval that an interactive diagnostic test will wait for a client to respond.

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

718 **7.5.4 CIM_JobSettingData.TerminateOnTimeout**

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

721 This property defines the behavior when a client fails to respond within the time interval defined by
722 CIM_JobSettingData.InteractiveTimeout. If this value is TRUE the job will terminate when the
723 InteractiveTimeout is exceeded on the last retry, if applicable. If FALSE the job will use
724 DefaultInputValues as the response and resume the test.

725 **7.5.5 CIM_JobSettingData.DefaultInputValues**

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

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

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

735 **7.5.6 CIM_JobSettingData.DefaultInputNames**

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

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

743 **7.5.7 CIM_JobSettingData.ClientRetries**

744 This property shall have a value if the CIM_DiagnosticTest.Characteristics has the value of 3 (Is
745 Interactive). Otherwise, this property value is ignored.

746 This integer property indicates the number of times the diagnostic test will prompt the client for the same
747 response after the client fails to invoke the CIM_ConcreteJob.ResumeWithInput() or
748 CIM_ConcreteJob.ResumeWithAction() extrinsic method within the specified
749 CIM_JobSettingData.InteractiveTimeout period of time . A value of one for this property indicates that the
750 diagnostic test will issue a second prompt for the same response.

751 **7.5.8 CIM_JobSettingData.RunInSilentMode**

752 This boolean property indicates whether the diagnostic test will not prompt the client for responses even
753 though CIM_DiagnosticTest.Characteristics contains the value of 3 (Is Interactive). When the value is
754 TRUE, no prompts are issued. Instead, the diagnostic test will execute using the values defined in default

755 CIM_JobSettingData.DefaultInputValues or the JobSettings parameter of the RunDiagnosticService.
756 When the value is FALSE, the interactive diagnostic test will prompt the client for a response.

757 If CIM_DiagnosticServiceJobCapabilities.SilentModeSupported has the value of FALSE, this property is
758 ignored.

759 **EXPERIMENTAL**

760 **7.6 Interactive options**

761 An interactive diagnostic test is controlled by properties in the optional
762 CIM_DiagnosticServiceJobCapabilities class, the properties in the mandatory CIM_JobSettingData
763 (Default) class, and the JobSettings input parameter, which is an embedded instance of
764 CIM_JobSettingData(Client), used when the client invokes the
765 CIM_DiagnosticTest.RunDiagnosticService() extrinsic method.

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

- 769 • CIM_DiagnosticServiceJobCapabilities.InteractiveTimeoutMax
- 770 • CIM_DiagnosticServiceJobCapabilities.ClientRetriesMax
- 771 • CIM_JobSettingData.InteractiveTimeout
- 772 • CIM_JobSettingData.TerminateOnTimeout

773 When a timeout occurs or when running in silent mode, the diagnostic test may resume using default
774 values, CIM_JobSettingData.DefaultInputValues, for the arguments that the client should have provided.
775 How default values can be used is determined by the combination of values in the following properties:
776 Table 3 shows the behavior when the following properties and JobSettings parameter have certain and
777 possibly conflicting values.

- 778 • CIM_DiagnosticServiceJobCapabilities.DefaultValuesSupported
- 779 • CIM_JobSettingData.DefaultInputValues
- 780 • CIM_JobSettingData.DefaultInputNames

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

- 784 • CIM_DiagnosticServiceJobCapabilities.SilentModeSupported
- 785 • CIM_JobSettingData.RunInSilentMode

Table 3 – Interactive options

DiagnosticServiceJob Capabilities	JobSettingData (Default)	JobSettings [JobSettingData (Client)]	Behavior
InteractiveTimeoutMax	InteractiveTimeout <= InteractiveTimeoutMax	InteractiveTimeout <= InteractiveTimeoutMax	Use JobSettings value
		InteractiveTimeout >= InteractiveTimeoutMax	Use DiagnosticServiceJob Capabilities value (2)
DefaultValuesSupported = TRUE	DefaultInputValues = non-NULL DefaultInputNames = non-NULL	DefaultInputValues = non-NULL DefaultInputNames = non-NULL	Use JobSettings
		DefaultInputValues = NULL DefaultInputNames = NULL	Use JobSettingData (Default).(2)
DefaultValuesSupported = FALSE	DefaultInputValues = NULL DefaultInputNames = NULL	DefaultInputValues = NULL DefaultInputNames = NULL	The client shall supply input values.
		DefaultInputValues = non-NULL DefaultInputNames = non-NULL	JobSettings is ignored. (3) The client shall supply input values in response to DIAG34.
DefaultValuesSupported = TRUE SilentModeSupported = TRUE	DefaultInputValues = non-NULL DefaultInputNames = non-NULL	DefaultInputValues = non-NULL DefaultInputNames = non-NULL RunInSilentMode = TRUE	Run in silent mode using JobSettings default values.
	DefaultInputValues = non-NULL DefaultInputNames = non-NULL	DefaultInputValues = NULL DefaultInputNames = 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. TerminateOnTimeout (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)
ClientRetriesMax = N	ClientRetries = M where M <= N	ClientRetries = R where R > N	After M retries, the job terminates or runs with defaults. (1) (2)

DiagnosticServiceJob Capabilities	JobSettingData (Default)	JobSettings [JobSettingData (Client)]	Behavior
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 ClientRetriesMax = N	TerminateOnTimeout = FALSE		Undefined
ClientRetriesMax = N	ClientRetries = M where M > N		Undefined

- 787 (1) The job will wait one InteractiveTimeout for each response. After the timeout, another prompt is
788 issued.
- 789 (2) An alert indication (DIAG39) is sent indicating that the JobSettings, which is an embedded
790 instance of CIM_JobSettingData(Client), was reset. The overridden effective JobSettings should
791 be logged.
- 792 (3) An alert indication (DIAG40) is sent indicating that the default values were not used. The
793 overridden effective JobSettings should be logged.

794 **7.7 Job deletion options**

795 To start a diagnostic test, the client invokes the CIM_DiagnosticTest.RunDiagnosticService extrinsic
796 method, which returns an instance of CIM_ConcreteJob. After a diagnostic test is completed, the
797 CIM_ConcreteJob instance will be deleted. When and how the CIM_ConcreteJob instance is deleted is
798 controlled by properties in the optional CIM_DiagnosticServiceJobCapabilities class, the properties in the
799 mandatory CIM_JobSettingData (Default) class, and the optional JobSettings input parameter, which is
800 an embedded CIM_JobSettingData instance, used when the client invokes the
801 CIM_DiagnosticTest.RunDiagnosticService() extrinsic method. Table 4 shows the behavior when these
802 following properties have certain and possibly conflicting values.

- 803 • CIM_DiagnosticServiceJobCapabilities.DeleteJobSupported
- 804 • CIM_DiagnosticServiceJobCapabilities.CleanupInterval
- 805 • CIM_JobSettingData.DeleteOnCompletion
- 806 • CIM_JobSettingData.TimeBeforeRemoval

807 Since the JobSettings input parameter, which is an embedded instance of CIM_JobSettingData (Client),
808 is optional, its value may be NULL. In this case, the CIM_DiagnosticTest.RunDiagnosticService extrinsic
809 method shall use the values of the mandatory CIM_JobSettingData (Default) instance.

810 An instance of CIM_DiagnosticServiceJobCapabilities may not exist because its implementation is
811 optional. In this case, the other class properties present will dictate behavior.

812

Table 4 – Job deletion options

DiagnosticServiceJob Capabilities	JobSettingData (Default)	JobSettings [JobSettingData (Client)]	Behavior
DeleteJobSupported = TRUE AND CleanupInterval = non-NULL	DeleteOnCompletion = TRUE or FALSE	DeleteOnCompletion = TRUE	The provider deletes the job instance after the job is completed and an elapsed time of TimeBeforeRemoval
		DeleteOnCompletion = FALSE	The provider will delete the job instance after the job is completed should the client fail to do so within an elapsed time of CleanupInterval
DeleteJobSupported = TRUE AND CleanupInterval = NULL	DeleteOnCompletion = TRUE	DeleteOnCompletion = TRUE	The provider deletes the job instance after the job is completed and an elapsed time of TimeBeforeRemoval
		DeleteOnCompletion = FALSE	The client should delete the job instance after the job is completed. The provider will not automatically delete the job. The CleanupInterval is ignored.
DeleteJobSupported = FALSE AND CleanupInterval = non-NULL	DeleteOnCompletion = TRUE	DeleteOnCompletion = TRUE	The provider deletes the job instance after the job is completed and an elapsed time of TimeBeforeRemoval
		DeleteOnCompletion = FALSE	JobSetting.DeleteOnCompletion is reset to TRUE. The provider deletes the job instance after the job is completed and an elapsed time of TimeBeforeRemoval. (1)
DeleteJobSupported = FALSE AND CleanupInterval = NULL	DeleteOnCompletion = FALSE		Undefined
DeleteJobSupported = FALSE AND CleanupInterval = non-NULL	DeleteOnCompletion = FALSE		Undefined

813
814

(1) An alert indication (DIAG39) is sent indicating that the JobSettings, which is an embedded instance of CIM_JobSettingData (Client), was reset. The overridden effective JobSettings should be logged.

815 **7.8 Diagnostic Job Control Profile indications support**

816 The Diagnostic Job Control Profile constrains certain elements in its support for the DMTF Indications
817 Profile. This subclause identifies those constraints.

818 **7.8.1 CIM_IndicationFilter (StaticIndicationFilter)**

819 The Diagnostic Job Control Profile constrains some of the properties of the StaticIndicationFilter version
820 of the CIM_IndicationFilter class and makes the class mandatory. The class is mandatory because some
821 of the alert indication filters are mandatory and the Diagnostic Job Control profile requires that static
822 versions of mandatory indication filters be populated.

823 **7.8.1.1 CIM_IndicationFilter.Name**

824 The Diagnostic Job Control Profile constrains names of the profile defined alert indication filters as
825 prescribed by [DSP1054](#). The names for the indication filters are identified in the entries for the indications
826 in Table 15. The Name shall be formatted as defined by the following ABNF rule:

827 "DMTF:Diagnostic Job Control:" MessageID

828 The MessageID shall have the same value of the MessageID in the Query for the filter.

829 **7.8.1.2 CIM_IndicationFilter.Query**

830 The Diagnostic Job Control Profile constrains the Query properties of the profile defined alert indication
831 filters as prescribed by [DSP1054](#). The Query properties for the indication filters are identified in the
832 entries for the indications in Table 15.

833 **7.8.1.3 CIM_IndicationFilter.QueryLanguage**

834 The Diagnostic Job Control Profile constrains the QueryLanguage properties of the profile defined alert
835 indication filters as prescribed by [DSP1054](#). The QueryLanguage properties for the indication filters are
836 identified in the entries for the indications in Table 15.

837 **7.8.2 CIM_FilterCollection (ProfileSpecificFilterCollection)**

838 The Diagnostic Job Control Profile constrains the CollectionName property of the
839 ProfileSpecificFilterCollection version of the CIM_FilterCollection class.

840 **7.8.2.1 CIM_FilterCollection.CollectionName**

841 The Diagnostic Job Control Profile constrains CollectionName of the profile defined
842 ProfileSpecificFilterCollection filter collection as prescribed by [DSP1054](#). The CollectionName for the filter
843 collection shall be formatted as defined by the following ABNF rule:

844 "DMTF:Diagnostic Job Control:ProfileSpecifiedAlertIndicationFilterCollection"

845 **7.8.3 CIM_MemberOfCollection (IndicationFilterInFilterCollection)**

846 **7.8.3.1 CIM_MemberOfCollection.Collection**

847 The Diagnostic Job Control Profile constrains the Collection property to be the reference to the
848 ProfileSpecificFilterCollection filter collection.

849 **7.8.3.2 CIM_MemberOfCollection.Member**

850 The Diagnostic Job Control Profile constrains the Member property to be a reference to one of the profile
851 defined alert indication filters.

852 7.8.4 CIM_OwningCollectionElement (IndicationServiceOfFilterCollection)

853 7.8.4.1 CIM_OwningCollectionElement.OwnedElement

854 The Diagnostic Job Control Profile constrains OwnedElement property to be the reference to the
855 ProfileSpecifiedFilterCollection filter collection.

856 7.9 Diagnostics job control alert indications and standard messages

857 7.9.1 DIAG9 – Test continued after last interactive timeout using default values

858 The interactive test experienced a timeout on its last query to the user and was resumed using default
859 values.

860 This alert would only be sent if the test job was interactive (CIM_DiagnosticTest.Characteristics includes
861 “3” (Is Interactive). The alert is sent when the JobSettings.TerminateOnTimeout = FALSE and the
862 JobSettings.InteractiveTimeout and JobSettings.ClientRetries are exceeded before the client provides a
863 response to an alert for either input or action (see DIAG34 in subclause 7.9.6 and DIAG35 in subclause
864 7.9.7). When the JobSettings.TerminateOnTimeout = FALSE and the JobSettings.ClientRetries have
865 been exceeded, the test resumes with Default Values.

866 The variables in this message are:

- 867 • Diagnostic Test Name – Identifies the Diagnostic Test instance that was run. This is the Name
868 property of the DiagnosticTest instance.
- 869 • Element Moniker – Identifies a unique name for the element under test (such as, Disk Drive)
870 that was specified.

871 This could be one of the following:

- 872 – The Object path of the element
- 873 – The ElementName of the element
- 874 – A unique user friendly name not in the model (such as, asset name)

875 The Element Moniker can be any of these, but whichever one is used shall be used consistently
876 for all managed elements of the same type within the scoping profile (such as, all disk drives in
877 a system).

878 With this alert the AlertType shall have the value 1 (Other). The OtherAlertType should be set to “Default
879 Values Used”.

880 With this alert the PerceivedSeverity shall have the value 3 (Warning).

881 7.9.2 DIAG12 – Job could not be started

882 The test job could not be started. The test was not run.

883 This alert would be sent if conditions are such that the test could not be executed. This message shall be
884 sent if the ReturnCode on RunDiagnosticService is non-zero. Some of the reasons that this might be true
885 are:

- 886 • Element already under test. Too many jobs are currently running
- 887 • The test is disabled
- 888 • The element is disabled
- 889 • The element under test is in recovery
- 890 • Resources are inadequate to run job

891 The variables in this message are:

892 • Diagnostic Test Name – Identifies the Diagnostic Test instance that was run. This is the Name
893 property of the DiagnosticTest instance.

894 • Element Moniker – Identifies a unique name for the element under test (such as, Disk Drive)
895 that was specified.

896 This could be one of the following:

- 897 – The Object path of the element
- 898 – The ElementName of the element
- 899 – A unique user friendly name not in the model (such as, asset name)

900 The Element Moniker can be any of these, but whichever one is used shall be used consistently
901 for all managed elements of the same type within the scoping profile (e.g., all disk drives in a
902 system).

903 • Reason – Identifies the reason the job was not started.

904 With this alert the AlertType shall have the value 4 (Processing Error)

905 With this alert the PerceivedSeverity shall have the value 5 (Major).

906 **7.9.3 DIAG19 – Test killed by client**

907 The test was killed by the client using the RequestedStateChange method.

908 This alert would be sent if the client issued a RequestedStateChange method on the ConcreteJob for the
909 test and the change was successfully executed.

910 NOTE If the RequestedStateChange failed for any reason, then this is indicated to the client that issued the
911 request with a ReturnCode on the RequestedStateChange. No alert message is sent. The client should retry or take
912 the appropriate remedial action on the test.

913 The variables in this message are:

914 • Diagnostic Test Name – Identifies the Diagnostic Test instance that was run. This is the Name
915 property of the DiagnosticTest instance.

916 • Element Moniker – Identifies a unique name for the element under test (such as, Disk Drive)
917 that was specified.

918 This could be one of the following:

- 919 – The Object path of the element
- 920 – The ElementName of the element
- 921 – A unique user friendly name not in the model (such as, asset name)

922 The Element Moniker can be any of these, but whichever one is used shall be used consistently
923 for all managed elements of the same type within the scoping profile (such as, all disk drives in
924 a system).

925 With this alert the AlertType shall have the value 1 (Other). OtherAlertType should be set to “Killed by
926 Client”.

927 With this alert the PerceivedSeverity shall have the value 2 (Information).

928 **7.9.4 DIAG20 – Test terminated by client**

929 The test was terminated by the client using the RequestedStateChange method.

930 This alert would be sent if the client issued a RequestedStateChange method on the ConcreteJob for the
931 test and the change was successfully executed.

932 NOTE If the RequestedStateChange failed for any reason, then this is indicated to the client that issued the
933 request with a ReturnCode on the RequestedStateChange. No alert message is sent. The client should retry or take
934 the appropriate remedial action on the test.

935 The variables in this message are:

- 936 • Diagnostic Test Name – Identifies the Diagnostic Test instance that was run. This is the Name
937 property of the DiagnosticTest instance.
- 938 • Element Moniker – Identifies a unique name for the element under test (such as, Disk Drive)
939 that was specified.

940 This could be one of the following:

- 941 – The Object path of the element
- 942 – The ElementName of the element
- 943 – A unique user friendly name not in the model (such as, asset name)

944 The Element Moniker can be any of these, but whichever one is used shall be used consistently
945 for all managed elements of the same type within the scoping profile (such as, all disk drives in
946 a system).

947 With this alert the AlertType shall have the value 1 (Other). OtherAlertType should be set to “Terminated
948 by Client”.

949 With this alert the PerceivedSeverity shall have the value 2 (Information).

950 **7.9.5 DIAG21 – Test suspended by client**

951 The test was suspended by a client that issued a RequestedStateChange setting the new state to
952 suspended.

953 This alert would be sent if a client issues a RequestedStateChange on the ConcreteJob specifying the
954 new state as Suspended, the implementation supports the state change and the implementation
955 successfully executes the request.

956 NOTE If the RequestedStateChange failed for any reason, then this is indicated to the client that issued the
957 request with a ReturnCode on the RequestedStateChange. No alert message is sent. The client should retry or take
958 the appropriate remedial action on the test.

959 The variables in this message are:

- 960 • Diagnostic Test Name – Identifies the Diagnostic Test instance that was run. This is the Name
961 property of the DiagnosticTest instance.
- 962 • Element Moniker – Identifies a unique name for the element under test (such as, Disk Drive)
963 that was specified.

964 This could be one of the following:

- 965 – The Object path of the element
- 966 – The ElementName of the element
- 967 – A unique user friendly name not in the model (such as, asset name)

968 The Element Moniker can be any of these, but whichever one is used shall be used consistently
969 for all managed elements of the same type within the scoping profile (such as, all disk drives in
970 a system).

971 With this alert the AlertType shall have the value 1 (Other). The OtherAlertType should be set to
972 “Suspended by Client”

973 With this alert the PerceivedSeverity shall have the value 2 (Information).

974 7.9.6 DIAG34 – Request for inputs

975 This is an alert indication to solicit input to an interactive test from a client.

976 This alert would be sent if the test is interactive and requires input values to continue running when
977 SilentMode is set to false.

978 The variables in this message are:

- 979 • Diagnostic Test Name – Identifies the Diagnostic Test instance that was run. This is the Name
980 property of the DiagnosticTest instance.
- 981 • Element Moniker – Identifies a unique name for the element under test (e.g., Disk Drive) that
982 was specified.

983 This could be one of the following:

- 984 – The Object path of the element
- 985 – The ElementName of the element
- 986 – A unique user friendly name not in the model (such as, asset name)

987 The Element Moniker can be any of these, but whichever one is used shall be used consistently
988 for all managed elements of the same type within the scoping profile (such as, all disk drives in
989 a system).

- 990 • InputNames – Identifies the names of values that the test is requesting.

991 With this alert the AlertType shall have the value 1 (Other). OtherAlertType should be set to “Request for
992 Input”.

993 With this alert the PerceivedSeverity shall have the value 2 (Information).

994 7.9.7 DIAG35 – Request for action

995 This is an alert indication to solicit user action from a client on an interactive test.

996 This alert would be sent if the test is interactive and requires user actions to continue running when
997 SilentMode is set to false.

998 The variables in this message are:

- 999 • Action String - Identifies the action being requested
- 1000 • Diagnostic Test Name – Identifies the Diagnostic Test instance that was run. This is the Name
1001 property of the DiagnosticTest instance.
- 1002 • Element Moniker – Identifies a unique name for the element under test (such as, Disk Drive)
1003 that was specified.

1004 This could be one of the following:

- 1005 – The Object path of the element
- 1006 – The ElementName of the element
- 1007 – A unique user friendly name not in the model (such as, asset name)

1008 The Element Moniker can be any of these, but whichever one is used shall be used consistently
1009 for all managed elements of the same type within the scoping profile (such as, all disk drives in
1010 a system).

1011 With this alert the AlertType shall have the value 1 (Other). OtherAlertType should be set to “Request for
1012 Action”.

1013 With this alert the PerceivedSeverity shall have the value 2 (Information).

1014 **7.9.8 DIAG36 – Test killed by test**

1015 The test killed itself. The test was killed and limited or no clean up was done.

1016 This alert would be sent if the test could not complete due to some error. There may be other (earlier)
1017 alerts that specify the specific error. Note this could happen as a result of subtests needed in executing
1018 the test. This alert may also be sent if the test that was killed was a subtest of a parent test.

1019 **NOTE** With this alert message, the client may need to take action to effect clean up in order to rerun the test.

1020 The variables in this message are:

- 1021 • Diagnostic Test Name – Identifies the Diagnostic Test instance (possibly a subtest) that was
1022 killed. This is the Name property of the DiagnosticTest instance.
- 1023 • Element Moniker – Identifies a unique name for the element under test (such as, Disk Drive)
1024 that was killed.

1025 This could be one of the following:

- 1026 – The Object path of the element
- 1027 – The ElementName of the element
- 1028 – A unique user friendly name not in the model (such as, asset name)

1029 The Element Moniker can be any of these, but whichever one is used shall be used consistently
1030 for all managed elements of the same type within the scoping profile (such as, all disk drives in
1031 a system).

1032 With this alert the AlertType shall have the value 4 (Processing Error). It would be a Processing Error if
1033 there was a test software error or a fault with the device that caused the test to be killed.

1034 With this alert the PerceivedSeverity shall have the value 5 (Major).

1035 **7.9.9 DIAG37 – Test terminated by test**

1036 The test terminated itself. The test was terminated and clean up was done.

1037 This alert would be sent if the test could not complete and some error occurred. There may be other
1038 (earlier) alerts that specify the specific error. Note this could happen as a result of subtests needed to
1039 executing the test. This alert may also be sent if the test that was terminated was a subtest of a parent
1040 test.

1041 **NOTE** With this alert message, the client need not take action to effect clean up in order to rerun the test.

1042 The variables in this message are:

- 1043 • Diagnostic Test Name – Identifies the Diagnostic Test instance that was terminated. This is the
1044 Name property of the DiagnosticTest instance.
- 1045 • Element Moniker – Identifies a unique name for the element under test (such as, Disk Drive)
1046 that was specified.

1047 This could be one of the following:

- 1048 – The Object path of the element
- 1049 – The ElementName of the element
- 1050 – A unique user friendly name not in the model (such as, asset name)

1051 The Element Moniker can be any of these, but whichever one is used shall be used consistently
1052 for all managed elements of the same type within the scoping profile (such as, all disk drives in
1053 a system).

1054 With this alert the AlertType shall have the value 4 (Processing Error). It would be a Processing Error if
1055 there was a test software error or a fault with the device that caused the test to be terminated, since it
1056 does not imply there is anything wrong with the device.

1057 With this alert the PerceivedSeverity shall have the value 5 (Major).

1058 **7.9.10 DIAG38 – Test resumed by client**

1059 The suspended test was resumed by a client that issued a RequestedStateChange setting the new state
1060 to start.

1061 This alert would be sent if a client issues a RequestedStateChange on the ConcreteJob specifying the
1062 new state as Start when the current job state is Suspended, the implementation supports the state
1063 change, and the implementation successfully executes the request.

1064 NOTE If the RequestedStateChange failed for any reason, then this is indicated to the client that issued the
1065 request with a ReturnCode on the RequestedStateChange. No alert message is sent. The client should retry or take
1066 the appropriate remedial action on the test.

1067 The variables in this message are:

- 1068 • Diagnostic Test Name – Identifies the Diagnostic Test instance that was run. This is the Name
1069 property of the DiagnosticTest instance.
- 1070 • Element Moniker – Identifies a unique name for the element under test (such as, Disk Drive)
1071 that was specified.

1072 This could be one of the following:

- 1073 – The Object path of the element
- 1074 – The ElementName of the element
- 1075 – A unique user friendly name not in the model (such as, asset name)

1076 The Element Moniker can be any of these, but whichever one is used shall be used consistently
1077 for all managed elements of the same type within the scoping profile (such as, all disk drives in
1078 a system).

1079 With this alert the AlertType shall have the value 1 (Other). The OtherAlertType should be set to “Resume
1080 Requested”.

1081 With this alert the PerceivedSeverity shall have the value 2 (Information).

1082 **7.9.11 DIAG39 – JobSettings reset**

1083 The test was run with the specified JobSettings parameter on RunDiagnosticService reset to match what
1084 the test is capable of supporting.

1085 This alert would be sent if a JobSettings supplied on the invocation of RunDiagnosticService cannot be
1086 supported and an element of the supplied JobSettingData is replaced. The Alert message identifies the
1087 JobSettings property that was reset and the value it was reset to. If multiple JobSettings properties are
1088 reset then there would be multiple instances of this alert (each identifying one property that was reset).

1089 The variables in this message are:

- 1090 • Diagnostic Test Name – Identifies the Diagnostic Test instance that was run. This is the Name
1091 property of the DiagnosticTest instance.

- 1092 • Element Moniker – Identifies a unique name for the element under test (such as, Disk Drive)
1093 that was specified.

1094 This could be one of the following:

- 1095 – The Object path of the element
- 1096 – The ElementName of the element
- 1097 – A unique user friendly name not in the model (such as, asset name)

1098 The Element Moniker can be any of these, but whichever one is used shall be used consistently
1099 for all managed elements of the same type within the scoping profile (such as, all disk drives in
1100 a system).

- 1101 • JobSettings Property – Identifies the JobSettings property that was reset.
- 1102 • JobSettings Value – Identifies the JobSettings property value that was used

1103 With this alert the AlertType shall have the value 1 (Other). The OtherAlertType should be set to
1104 “JobSettings Reset”.

1105 With this alert the PerceivedSeverity shall have the value 3 (Warning).

1106 For more information on this alert see subclause 7.6 and note 2 for entries in the table.

1107 **7.9.12 DIAG40 – JobSettings defaults not used**

1108 The test ran, but the default values for interactive input as specified in the JobSettings parameter were
1109 not used.

1110 This alert would be sent if the client provides default values in JobSettings on invocation of the
1111 RunDiagnosticService method but the implementation does not support default values as input (as
1112 defined in the DiagnosticServiceCapabilities.DefaultValuesSupported property).

1113 The variables in this message are:

- 1114 • Diagnostic Test Name – Identifies the Diagnostic Test instance that was run. This is the Name
1115 property of the DiagnosticTest instance.
- 1116 • Element Moniker – Identifies a unique name for the element under test (such as, Disk Drive)
1117 that was specified.

1118 This could be one of the following:

- 1119 – The Object path of the element
- 1120 – The ElementName of the element
- 1121 – A unique user friendly name not in the model (such as, asset name)

1122 The Element Moniker can be any of these, but whichever one is used shall be used consistently
1123 for all managed elements of the same type within the scoping profile (such as, all disk drives in
1124 a system).

1125 With this alert the AlertType shall have the value 1 (Other). The OtherAlertType should be set to “Defaults
1126 Not Used”.

1127 With this alert the PerceivedSeverity shall have the value 3 (Warning).

1128 For more information on this alert see subclause 7.6 and note 3 for entries in the table.

1129 7.9.13 DIAG48 – Test continued after an interim interactive timeout

1130 The interactive test experienced a timeout on one of its queries (but not the last) to the user. The test re-
1131 issued the query for inputs or actions because the number of retries has not been exhausted.

1132 This alert would only be sent if the test job was interactive (CIM_DiagnosticTest.Characteristics includes
1133 “3” (Is Interactive) and the JobSettings.InteractiveTimeout is exceeded before the client provides a
1134 response to an alert for either input or action (see DIAG34 in subclause 7.9.6 and DIAG35 in subclause
1135 7.9.7). If JobSettings.TerminateOnTimeout = FALSE was specified and the JobSettings.ClientRetries
1136 have not been exceeded, the test will reissue the query (DIAG34 or DIAG35).

1137 A separate test completion status message will be sent later.

1138 The variables in this message are:

- 1139 • Diagnostic Test Name – Identifies the Diagnostic Test instance that was run. This is the Name
1140 property of the DiagnosticTest instance.
- 1141 • Element Moniker – Identifies a unique name for the element under test (such as, Disk Drive)
1142 that was specified.

1143 This could be one of the following:

- 1144 – The Object path of the element
- 1145 – The ElementName of the element
- 1146 – A unique user friendly name not in the model (such as, asset name)

1147 The Element Moniker can be any of these, but whichever one is used shall be used consistently
1148 for all managed elements of the same type within the scoping profile (such as, all disk drives in
1149 a system).

1150 With this alert the AlertType shall have the value 1 (Other). The OtherAlertType should be set to “Interim
1151 Interactive Timeout”.

1152 With this alert the PerceivedSeverity shall have the value 2 (Information).

1153 7.9.14 DIAG49 – Test terminated after an interactive timeout

1154 The interactive test experienced a timeout on one of its queries to the user. The test execution is
1155 terminated because JobSettings.TerminateOnTimeout was set to TRUE and the number of retries has
1156 been exhausted.

1157 This alert would only be sent if the test job was interactive (CIM_DiagnosticTest.Characteristics includes
1158 “3” (Is Interactive) and the JobSettings.InteractiveTimeout is exceeded before the client provides a
1159 response to an alert for either input or action (see DIAG34 in subclause 7.9.6 and DIAG35 in subclause
1160 7.9.7). If JobSettings.TerminateOnTimeout = TRUE was specified on the request, then the Job (and test)
1161 is terminated with a JobState of 8 (Terminated).

1162 A separate test completion status message (DIAG45) will be sent later.

1163 The variables in this message are:

- 1164 • Diagnostic Test Name – Identifies the Diagnostic Test instance that was run. This is the Name
1165 property of the DiagnosticTest instance.
- 1166 • Element Moniker – Identifies a unique name for the element under test (such as, Disk Drive)
1167 that was specified.

- 1168 This could be one of the following:
- 1169 – The Object path of the element
 - 1170 – The ElementName of the element
 - 1171 – A unique user friendly name not in the model (such as, asset name)
- 1172 The Element Moniker can be any of these, but whichever one is used shall be used consistently
1173 for all managed elements of the same type within the scoping profile (such as, all disk drives in
1174 a system).
- 1175 With this alert the AlertType shall have the value 4 (Processing Error). The test job is terminated.
- 1176 With this alert the PerceivedSeverity shall have the value 5 (Major).

1177 8 Methods

1178 This clause details the requirements for supporting intrinsic operations and extrinsic methods for the CIM
1179 elements defined by this profile.

1180 8.1 Profile conventions for operations

- 1181 The default list of operations shall be as mandated in [DSP1103](#), subclauses 8.1.
- 1182 Support for operations for each profile class (including associations) shall be as mandated in [DSP1103](#),
1183 subclauses 8.3 through 8.7.

1184 8.2 CIM_ConcreteJob

1185 All operations are supported as for CIM_ConcreteJob in [DSP1103](#), subclause 8.2. Additionally, the
1186 DeleteInstance operation shall be supported when the CIM_JobSettingData.DeleteOnCompletion
1187 property has the value of FALSE.

1188 8.2.1 CIM_ConcreteJob.RequestStateChange()

- 1189 All successful CIM_DiagnosticService.RunDiagnosticService() calls will return a reference to a
1190 CIM_ConcreteJob instance, which represents the diagnostic execution. The
1191 CIM_ConcreteJob.RequestStateChange() method is invoked to control the diagnostic program execution.
1192 The RequestedState input parameter specifies the new desired state (Start, Suspend, Kill, Terminate).
- 1193 Otherwise, the behavior of this extrinsic method shall be as mandated in [DSP1103](#), subclause 8.2.
- 1194 Before invoking this method, a client examines
1195 CIM_DiagnosticServiceJobCapabilities.RequestedStatesSupported to determine the values to use for the
1196 RequestedState input parameter. The RequestStateChange() extrinsic method shall change the JobState
1197 value if the transition is successfully performed.
- 1198 If the RequestedStateChange is successful (ReturnCode = 0), the
1199 CIM_ConcreteJob.TimeOfLastStateChange shall be set to the date and time of the successful
1200 completion.

1201 EXPERIMENTAL

- 1202 If the RequestedStateChange is successful (ReturnCode = 0) and the client has subscribed to the
1203 following indications, then the following AlertIndications will be sent to the client (depending on the state
1204 change requested):

- 1205 • DIAG19 – Test Killed by client (see subclause 7.9.3)
- 1206 • DIAG20 – Test Terminated by client (see subclause 7.9.4)
- 1207 • DIAG21 – Test Suspended by client (see subclause 7.9.5)
- 1208 • DIAG38 - Test Resumed by client (see subclause 7.9.10)

1209 8.2.2 CIM_ConcreteJob.ResumeWithInput()

1210 The CIM_ConcreteJob.ResumeWithInput() extrinsic method is invoked to resume the diagnostic program
1211 execution when it has a JobState of 12 (Query Pending).

1212 The return values are specified in Table 5. The input parameter is specified in Table 6. No output
1213 parameters are defined. No standard messages are defined.

1214 For an interactive test, the CIM_ConcreteJob provider prompts the client to respond with DIAG34
1215 message (see 7.9.6), which is a comma separated string of argument names. The client calls
1216 CIM_ConcreteJob.ResumeWithInput() to respond with values in the Inputs string array. The first value in
1217 the Inputs string array corresponds to the first argument in the DIAG34 message, and so on

1218 The Inputs string array shall have a value for each requested argument. A NULL value shall not be used.
1219 Instead the client can use a default value from CIM_JobSettingData (Default). To use a default value for a
1220 requested argument, the client looks for a matching name in CIM_JobSettingData.DefaultInputNames. If
1221 found, the client uses the corresponding value from CIM_JobSettingData.DefaultInputValues; that is, at
1222 the same array index.

1223 If the client invokes the CIM_ConcreteJob.ResumeWithInput() extrinsic method where the Inputs string
1224 array argument has invalid values or not enough values, then the CIM_ConcreteJob provider returns 5
1225 (Invalid Parameter) and sets CIM_ConcreteJob.JobState = 10 (Exception) provided no more retries
1226 remain.

1227 If CIM_ConcreteJob.ResumeWithInput() returns value=3 (Can NOT complete within Timeout Period), the
1228 client can retry provided more retries remain.

1229 If this method is supported, then DIAG34 (see 7.9.6) shall also be supported. In addition, the following
1230 alert messages may need to be supported:

- 1231 • DIAG9 (Test continued after last interactive timeout using Default Values) – This shall be
1232 supported if CIM_DiagnosticServiceJobCapabilities.DefaultValuesSupported has the value of
1233 TRUE.
- 1234 • DIAG48 (Test continued after an interim interactive timeout) – This shall be supported if
1235 CIM_DiagnosticServiceJobCapabilities.ClientRetriesMax has a value greater than one.
- 1236 • DIAG49 (Test terminated after an interactive test timeout) – This shall be supported if
1237 CIM_JobSettingData.TerminateOnTimeout has the value of TRUE.

1238 If the client is running an interactive test (CIM_DiagnosticTest.Characteristics includes “Is Interactive”),
1239 then the client should be subscribing for the interactive alert indications (such as, DIAG34). If the client
1240 does not subscribe to the interactive alert indications, it will never be notified that inputs are required and
1241 interactive timeouts will occur. Eventually the test will either run with default values or be terminated.

1242

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

1243

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).

1244 8.2.3 CIM_ConcreteJob.ResumeWithAction()

1245 The CIM_ConcreteJob.ResumeWithAction() extrinsic method is invoked to resume the diagnostic
1246 program execution when it has a JobState of 12 (Query Pending).

1247 The return values are specified in Table 7. No input or output parameters are defined. No standard
1248 messages are defined.

1249 For an interactive test, the CIM_ConcreteJob provider prompts the client to respond with DIAG35
1250 message. The client invokes the CIM_ConcreteJob.ResumeWithAction() extrinsic method when no
1251 arguments are requested. For example, the diagnostic test might prompt the user to attach the network
1252 cable before allowing the test to proceed.

1253 If this method is supported, then DIAG35 (see 7.9.7) shall also be supported. In addition, the following
1254 alert messages may need to be supported:

- 1255 • DIAG48 (Test continued after an interim interactive timeout) – This shall be supported if
1256 CIM_DiagnosticServiceJobCapabilities.ClientRetriesMax has a value greater than one.
- 1257 • DIAG49 (Test terminated after an interactive test timeout) – This shall be supported if
1258 CIM_JobSettingData.TerminateOnTimeout has the value of TRUE.

1259 If the client is running an interactive test (CIM_DiagnosticTest.Characteristics includes “Is Interactive”),
1260 then the client should be subscribing for the interactive alert indications (such as, DIAG35). If the client
1261 does not subscribe to the interactive alert indications, it will never be notified that an action required and
1262 interactive timeouts will occur. Eventually the test will be terminated.

1263

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 ResumeWithAction response was accepted, 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

1264 8.3 CIM_DiagnosticServiceJobCapabilities

1265 8.3.1 CreateGoalSettings()

1266 The CIM_DiagnosticServiceJobCapabilities.CreateGoalSettings() method, which is inherited from
1267 CIM_Capabilities, is invoked in the context of a specific CIM_DiagnosticServiceJobCapabilities instance.

1268 This method is used to create a CIM_JobSettingData instance using the
1269 CIM_DiagnosticServiceJobCapabilities as a template. The purpose of this method is to create a
1270 CIM_JobSettingData based on the CIM_DiagnosticServiceJobCapabilities on which this method is
1271 invoked and has properties set in line with those CIM_DiagnosticServiceJobCapabilities.

1272 CIM_DiagnosticServiceJobCapabilities is optional. If not specified the client can create JobSettings based
1273 on the CIM_JobSettingData (Default) instance. CIM_JobSettingData (Default) is mandatory and will be
1274 supplied by the provider.

1275 The return values are specified in Table 8. The parameters are specified in Table 9. No standard
1276 messages are defined.

1277

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

1278

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

1279 **EXPERIMENTAL**

1280 **8.4 CIM_MethodResult**

1281 All operations are supported as for CIM_MethodResult in [DSP1103](#).

1282 **8.5 CIM_OwningJobElement**

1283 All operations are supported as for CIM_OwningJobElement in [DSP1103](#).

1284 **8.6 CIM_AffectedJobElement**

1285 All operations are supported as for CIM_AffectedJobElement in [DSP1103](#).

1286 **8.7 CIM_AssociatedJobMethodResult**

1287 All operations are supported as for CIM_AssociatedJobMethodResult in [DSP1103](#).

1288 **8.8 CIM_HostedDependency**

1289 All operations are supported as for CIM_HostedDependency in [DSP1103](#).

1290 **8.9 CIM_RegisteredProfile**

1291 All operations are supported as for CIM_RegisteredProfile in [\(DSP1033\)](#).

1292 **8.10 CIM_JobSettingData**

1293 Table 10 lists operations that either have special requirements beyond those from [DSP0200](#) or shall not
 1294 be supported.

1295

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

1296 8.11 CIM_ElementSettingData

1297 Table 11 lists operations that either have special requirements beyond those from [DSP0200](#) or shall not
1298 be supported.

1299

Table 11 – Operations: CIM_ElementSettingData

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

1300 8.12 CIM_ElementCapabilities

1301 Table 12 lists operations that either have special requirements beyond those from [DSP0200](#) or shall not
1302 be supported.

1303

Table 12 – Operations: CIM_ElementCapabilities

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

1304 8.13 CIM_DiagnosticTest.RunDiagnosticService()

1305 [DSP1002](#) describes this extrinsic method. This subclause describes how the
1306 CIM_DiagnosticServiceJobCapabilities, the CIM_JobSettingData (Default), and the JobSettings input
1307 parameter affects the execution of this extrinsic method.

1308 The CIM_JobSettingData (Default) is mandatory. The CIM_DiagnosticServiceJobCapabilities and the
1309 JobSettings parameter of the RunDiagnosticService method are optional. If the
1310 CIM_DiagnosticServiceJobCapabilities is not implemented, the client application cannot alter the default
1311 CIM_JobSettingData for the diagnostic test and the JobSettings parameter should be NULL or set to the
1312 CIM_JobSettingData (Default). If the client application sets the JobSettings parameter to values that
1313 conflict with the default CIM_JobSettingData, the test will not fail, but the JobSettings parameter will be
1314 reset to the default values (the “effective” JobSettings) and a warning alert message (DIAG39, see
1315 7.9.11) will be issued. The effective JobSettings parameter values will also be logged in the associated
1316 CIM_DiagnosticLog.

1317 If CIM_DiagnosticServiceJobCapabilities is implemented, the client application may specify values in the
1318 JobSettings parameter that conform to the corresponding capability. For example, the client application
1319 may specify an InteractiveTimeout that is equal or less than the InteractiveTimeoutMax..If the client
1320 application specifies a value that is in conflict with the options allowed by the

1321 CIM_DiagnosticServiceJobCapabilities for the diagnostic test, the conflicting value will be reset to one of
 1322 two values: the value in the JobSettingData (Default) or the maximum allowed by the
 1323 CIM_DiagnosticServiceJobCapabilities. If the client invokes this extrinsic method and the JobSettings
 1324 parameter has the value of NULL, the CIM_JobSettingData (Default) will be used. In either case, if any
 1325 value was changed, an alert message (DIAG39, see 7.9.11) will be issued. Whether or not a value was
 1326 changed, the effective JobSettings used by the diagnostic test execution will be logged in the
 1327 CIM_DiagnosticLog.

1328 Table 13 shows the behavior for different combinations of CIM_DiagnosticServiceJobCapabilities,
 1329 CIM_JobSettingData (Default), and the JobSettings parameter.

1330 **Table 13 – Job settings options**

DiagnosticService JobCapabilities	JobSetting Data (Default)	JobSettings [CIM_JobSettingData (Client)]	Behavior
Absent	Present	NULL	Use JobSettingData (Default).
Absent	Present	No conflict in values exist	Use JobSettings.(Client input)
		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 JobSettings (Client input)
		Conflict in values exist	JobSettings (Client input) is modified to conform to the capabilities. One or more alerts are issued.
	Absent		Undefined.
Present	Present but conflicting		Undefined.

1331 The effective JobSettings used is logged.

1332 **EXPERIMENTAL**

1333 **9 Use cases (informative)**

1334 This clause contains use cases for the *Diagnostic Job Control Profile* that describes how a diagnostic test
 1335 behaves and interacts with a client. An interactive diagnostic test is a CIM_DiagnosticTest instance where
 1336 its Characteristics property contains the value 3 (Is Interactive).

1337 **9.1 Use case summary**

1338 Table 14 summarizes the use cases that are described in this clause. The use cases are categorized and
 1339 named, and references are provided to the subclause that describes the use case.

1340 **NOTE** Although use case names follow the convention for naming classes, properties, and methods in the
 1341 schema, this naming was done for readability only and does not imply any functionality attached to the name.

1342 The CIM_ prefix has been omitted from the class names in the use cases for readability.

1343

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.

1344 Before performing the use cases in this profile, it is assumed that a client has already utilized the use
 1345 case methodology defined in [DSP1002](#) to discover the DiagnosticTest instance.

1346 To start a test, the client invokes the DiagnosticTest.RunDiagnosticService() extrinsic method which
 1347 returns 0 (Success) and the object path of a ConcreteJob instance with ConcreteJob.JobState = 4
 1348 (Running) and CIM_ConcreteJob.PercentComplete = 0. Thereafter, ConcreteJob manages the diagnostic
 1349 test execution. Additionally, a client monitors and interacts with the diagnostic test via the returned
 1350 ConcreteJob instance.

1351 NOTE The DiagnosticTest.RunDiagnosticService() always returns a reference to a job when the method returns a
 1352 0 (Success). That is, there is no case where the test is executed without a job being created.

1353 NOTE An interactive diagnostic test may prompt a client more than once during test execution where some
 1354 prompts require user input while others do not.

1355 In the following examples, responses are enclosed in brackets:

1356 [Enter] indicates that the client pressed the Enter key, typically to select the default.

1357 <timeout> indicates that the client did not respond.

1358 9.2 User input required

1359 For an interactive test where user input is required, the ConcreteJob provider prompts the client to
 1360 respond with DIAG34 message, which is a comma separated string of argument names. The client calls
 1361 ConcreteJob.ResumeWithInput() to respond with values in the Inputs string array. The first value in the
 1362 Inputs string array corresponds to the first argument in the DIAG34 message, and so on.

1363 9.2.1 Single prompt and response has a valid value

1364 - *How many minutes do you want the test to run? [20] (Test starts)*

1365 1) Client calls DiagnosticTest.RunDiagnosticService() which returns 0 (Success) and the object
 1366 path of the ConcreteJob instance and starts the diagnostic test. ConcreteJob.JobState = 4
 1367 (Running) and ConcreteJob.PercentComplete = 0.

1368 2) ConcreteJob provider prompts client to respond with a DIAG34 message, which requests a value
 1369 for the Minutes argument.

1370 3) ConcreteJob provider sets ConcreteJob.JobState = 12 (Query Pending) to wait for client
 1371 response.

1372 4) Client calls ConcreteJob.ResumeWithInput() to respond with input argument Inputs[0]="20" for
 1373 the Minutes argument requested by DIAG34 message.

1374 5) ConcreteJob provider sets ConcreteJob.JobState = 4 (Running).

- 1375 6) ConcreteJob.ResumeWithInput () returns 0 (“Completed with No Error”) and test execution
1376 starts.
- 1377 7) After the test is completed successfully, the ConcreteJob provider sets ConcreteJob.JobState = 7
1378 (Completed), ConcreteJob.OperationalStatus= 17 (Completed) and ConcreteJob
1379 PercentComplete=100.

1380 9.2.2 Single prompt and response has multiple valid values

- 1381 - Which CPU speeds in GHz to you want to test? [2.4,3.0] (Test starts)
- 1382 1) Client calls DiagnosticTest.RunDiagnosticService() which returns 0 (Success) and the object
1383 path of the ConcreteJob instance and starts the diagnostic test. ConcreteJob.JobState = 4
1384 (Running) and ConcreteJob.PercentComplete = 0.
- 1385 2) ConcreteJob provider prompts client to respond with a DIAG34 message, which requests a value
1386 for the CPUSpeed1 and CPUSpeed2 arguments.
- 1387 3) ConcreteJob provider sets ConcreteJob.JobState = 12 (Query Pending) to wait for client
1388 response.
- 1389 4) Client calls ConcreteJob.ResumeWithInput() to respond with input argument Inputs[0]=”2.4” for
1390 the CPUSpeed1 argument and Inputs[1]=”3.0” for the CPUSpeed2 argument as requested by the
1391 DIAG34 message.
- 1392 5) ConcreteJob provider sets ConcreteJob.JobState = 4 (Running).
- 1393 6) ConcreteJob.ResumeWithInput () returns 0 (“Completed with No Error”) and test execution
1394 starts.
- 1395 7) After the test is completed successfully, the ConcreteJob provider sets ConcreteJob.JobState = 7
1396 (Completed), ConcreteJob.OperationalStatus= 17 (Completed) and ConcreteJob
1397 PercentComplete=100.

1398 9.2.3 Multiple prompts and responses required with partial test execution after each

- 1399 - Which network port do you want to test? [2] (Test execution starts)
- 1400 - Which network port do you want to test next? [3] (Test execution resumes)
- 1401 1) Client calls DiagnosticTest.RunDiagnosticService() which returns 0 (Success) and the object
1402 path of the ConcreteJob instance and starts the diagnostic test. ConcreteJob.JobState = 4
1403 (Running) and ConcreteJob.PercentComplete = 0.
- 1404 2) ConcreteJob provider prompts client to respond with a DIAG34 message, which requests a value
1405 for the Port1 argument.
- 1406 3) ConcreteJob provider sets ConcreteJob.JobState = 12 (Query Pending) to wait for client
1407 response.
- 1408 4) Client calls ConcreteJob.ResumeWithInput() to respond with input argument Inputs[0]=”2” for the
1409 Port1 argument requested by DIAG34 message.
- 1410 5) ConcreteJob provider sets ConcreteJob.JobState = 4 (Running) and
1411 ConcreteJob.PercentComplete = 25 and test execution starts.
- 1412 6) ConcreteJob.ResumeWithInput () returns 0 (“Completed with No Error”) and test execution
1413 resumes.
- 1414 7) Test execution finishes for port 2. ConcreteJob provider sets ConcreteJob.PercentComplete = 50
1415 and execution continues.
- 1416 8) ConcreteJob provider prompts client to respond with a DIAG34 message, which requests a value
1417 for the Port2 argument.

- 1418 9) ConcreteJob provider sets ConcreteJob.JobState = 12 (Query Pending) to wait for client
1419 response.
- 1420 10) Client calls ConcreteJob.ResumeWithInput() to respond with input argument Inputs[0]="3" for the
1421 Port2 argument requested by DIAG34 message.
- 1422 11) ConcreteJob provider sets ConcreteJob.JobState = 4 (Running) and
1423 ConcreteJob.PercentComplete = 50 and test execution starts.
- 1424 12) ConcreteJob.ResumeWithInput() returns 0 ("Completed with No Error") and test execution
1425 resumes.
- 1426 13) Test execution is completed for port 3, the last port. The test has now completed successfully, the
1427 ConcreteJob provider sets ConcreteJob.JobState = 7 (Completed) ,
1428 ConcreteJob.OperationalStatus= 17 (Completed) and ConcreteJob.PercentComplete=100.

1429 **9.2.4 Client does not respond to a prompt**

1430 - *Which network port do you want to test? <timeout>*

- 1431 1) Client calls DiagnosticTest.RunDiagnosticService() which returns 0 (Success) and the object
1432 path of the ConcreteJob instance and starts the diagnostic test. ConcreteJob.JobState = 4
1433 (Running) and ConcreteJob.PercentComplete = 0.
- 1434 2) ConcreteJob provider prompts client to respond with a DIAG34 message, which requests a value
1435 for the Port argument.
- 1436 3) ConcreteJob provider sets ConcreteJob.JobState = 12 (Query Pending) to wait for client
1437 response.
- 1438 4) When JobSettingData.InteractiveTimeout is exceeded, the ConcreteJob provider examines the
1439 value of JobSettingData.ClientRetriesMax.
- 1440 5) If the value of JobSettingData.ClientRetriesMax is zero, or if the number of retries has been
1441 exceeded, the ConcreteJob provider sets ConcreteJob.JobState = 10 (Exception) provided
1442 JobSettingData.TerminateOnTimeout=TRUE.
- 1443 6) If the value of JobSettingData.ClientRetriesMax is non-zero, steps 2, 3, 4, and 5 are repeated.

1444 NOTE If the value of JobSettingData.ClientRetriesMax is non-zero, the ConcreteJob provider monitors how many
1445 timeouts that have occurred.

1446 **9.2.5 Client responds with an invalid value**

1447 - *Which network port do you want to test? [-1]*

- 1448 • Client calls DiagnosticTest.RunDiagnosticService() which returns 0 (Success) and the object
1449 path of the ConcreteJob instance and starts the diagnostic test. ConcreteJob.JobState = 4
1450 (Running) and ConcreteJob.PercentComplete = 0.
- 1451 • ConcreteJob provider prompts client to respond with a DIAG34 message, which requests a value
1452 for the Port argument.
- 1453 • Client calls ConcreteJob.ResumeWithInput() to respond with input argument Inputs[0]="-1" for
1454 the Port1 argument requested by DIAG34 message.
- 1455 • The ConcreteJob provider detects that Inputs[0]="-1" is invalid.
- 1456 • ConcreteJob provider responds to ConcreteJob.ResumeWithInput with 5 (Invalid Parameter).
- 1457 • ConcreteJob provider sets ConcreteJob.JobState= 10 (Exception) if this was the last retry.
1458 Otherwise, the provider changes the JobState to 12 (Query Pending) and reissues the DIAG34

1459 **9.2.6 Client does not respond with enough valid values**

1460 - *Which CPU speeds in GHz do you want to test? [2.4]*

- 1461 1) Client calls DiagnosticTest.RunDiagnosticService() which returns 0 (Success) and the object
1462 path of the ConcreteJob instance and starts the diagnostic test. ConcreteJob.JobState = 4
1463 (Running) and ConcreteJob.PercentComplete = 0.
- 1464 2) ConcreteJob provider prompts client to respond with a DIAG34 message, which requests a value
1465 for the CPUSpeed1 and CPUSpeed2 arguments.
- 1466 3) Client calls ConcreteJob.ResumeWithInput() to respond with input argument Inputs[0]="2.4" for
1467 the CPUSpeed1 argument requested by DIAG34 message but does not provide a value for
1468 Inputs[1] for the CPUSpeed2 argument.
- 1469 4) The ConcreteJob provider detects that Inputs[1] has no value.
- 1470 5) ConcreteJob provider responds to ConcreteJob.ResumeWithInput with 5 (Invalid Parameter).
- 1471 6) ConcreteJob provider sets ConcreteJob.JobState= 10 (Exception) if this was the last retry.
1472 Otherwise, the provider changes the JobState to 12 (Query Pending) and reissues the DIAG34.

1473 9.3 User action required

1474 For an interactive diagnostic test where user action is required, the ConcreteJob provider prompts client
1475 to respond with a DIAG35 message. The client responds by invoking the
1476 ConcreteJob.ResumeWithAction() extrinsic method.

1477 9.3.1 Single prompt and response required

1478 - Press any key when the network cable has been attached. [Enter] (Test execution starts)

- 1479 1) Client calls DiagnosticTest.RunDiagnosticService() which returns 0 (Success) and the object
1480 path of the ConcreteJob instance and starts the diagnostic test. ConcreteJob.JobState = 4
1481 (Running) and ConcreteJob.PercentComplete = 0.
- 1482 2) ConcreteJob provider prompts client to respond with a DIAG35 message.
- 1483 3) ConcreteJob provider sets ConcreteJob.JobState = 12 (Query Pending) to wait for client
1484 response.
- 1485 4) Client calls ConcreteJob.ResumeWithAction() after attaching the cable.
- 1486 5) ConcreteJob provider sets ConcreteJob.JobState = 4 (Running).
- 1487 6) ConcreteJob.ResumeWithAction() returns 0 ("Completed with No Error") and test execution
1488 resumes.
- 1489 7) After the test is completed successfully, the ConcreteJob provider sets ConcreteJob.JobState = 7
1490 (Completed), ConcreteJob.OperationalStatus= 17 (Completed), and ConcreteJob
1491 PercentComplete=100.

1492 9.3.2 Multiple prompts and responses required before running the test

1493 - Press any key when the network cable has been attached. [Enter]

1494 - Press any key after the CD has been inserted into the drive. [Enter] (Test execution starts)

- 1495 1) Client calls DiagnosticTest.RunDiagnosticService() which returns 0 (Success) and the object
1496 path of the ConcreteJob instance and starts the diagnostic test. ConcreteJob.JobState = 4
1497 (Running) and ConcreteJob.PercentComplete = 0.
- 1498 2) ConcreteJob provider prompts client to respond with a DIAG35 message.
- 1499 3) ConcreteJob provider sets ConcreteJob.JobState = 12 (Query Pending) to wait for client
1500 response.
- 1501 4) Client calls ConcreteJob.ResumeWithAction() after attaching the cable.
- 1502 5) ConcreteJob provider sets ConcreteJob.JobState = 4 (Running).

- 1503 6) ConcreteJob.ResumeWithAction() returns 0 (“Completed with No Error”) and test execution
1504 resumes..
- 1505 7) ConcreteJob provider prompts client to respond with a DIAG35 message.
- 1506 8) ConcreteJob provider sets ConcreteJob.JobState = 12 (Query Pending) to wait for client
1507 response.
- 1508 9) Client calls ConcreteJob.ResumeWithAction() after inserting the CD into the drive,
1509 ConcreteJob.ResumeWithAction() returns 0 (“Completed with No Error”) and test execution
1510 resumes with the ConcreteJob.JobState set to 4 (Running).
- 1511 10) After test is completed successfully, ConcreteJob provider sets ConcreteJob.JobState = 7
1512 (Completed) and ConcreteJob PercentComplete=100.

1513 9.3.3 Multiple prompts and responses required with partial test execution after each

1514 - Press any key when the network cable has been attached. [Enter]. (Test execution starts)

1515 - Press any key when the LED is on. [Enter] (Test execution resumes)

1516 - Press any key when the LED is off. [Enter] (Test execution resumes)

- 1517 1) Client calls DiagnosticTest.RunDiagnosticService() which returns 0 (Success) and the object
1518 path of the ConcreteJob instance and starts the diagnostic test. ConcreteJob.JobState = 4
1519 (Running) and ConcreteJob.PercentComplete = 0.
- 1520 2) ConcreteJob provider prompts client to respond with a DIAG35 message.
- 1521 3) ConcreteJob provider sets ConcreteJob.JobState = 12 (Query Pending) to wait for client
1522 response.
- 1523 4) Client calls ConcreteJob.ResumeWithAction() after attaching the cable.
- 1524 5) ConcreteJob.ResumeWithAction() returns 0 (“Completed with No Error”) and test execution
1525 resumes with ConcreteJob.JobState set to 4 (Running) and ConcreteJob.PercentComplete set to
1526 25.
- 1527 6) ConcreteJob provider prompts client to respond with a DIAG35 message.
- 1528 7) ConcreteJob provider sets ConcreteJob.JobState = 12 (Query Pending) to wait for client
1529 response.
- 1530 8) Client calls ConcreteJob.ResumeWithAction() after verifying the LED is ON.
- 1531 9) ConcreteJob.ResumeWithAction() returns 0 (“Completed with No Error”) and test execution
1532 resumes with ConcreteJob.JobState set to 4 (Running) and ConcreteJob.PercentComplete set to
1533 50.
- 1534 10) ConcreteJob provider prompts client to respond with a DIAG35 message.
- 1535 11) ConcreteJob provider sets ConcreteJob.JobState = 12 (Query Pending) to wait for client
1536 response.
- 1537 12) Client calls ConcreteJob.ResumeWithAction() after verifying the LED is OFF.
- 1538 13) ConcreteJob.ResumeWithAction() returns 0 (“Completed with No Error”) and test execution
1539 resumes with ConcreteJob.JobState set to 4 (Running) and ConcreteJob.PercentComplete set to
1540 75.
- 1541 14) After test is completed successfully, ConcreteJob provider sets ConcreteJob.JobState = 7
1542 (Completed) and ConcreteJob PercentComplete=100.

1543 9.3.4 Client does not respond to a prompt

1544 - Press any key when the network cable has been attached. <timeout>

- 1545 1) Client calls DiagnosticTest.RunDiagnosticService() which returns 0 (Success) and the object
1546 path of the ConcreteJob instance and starts the diagnostic test. ConcreteJob.JobState = 4
1547 (Running) and ConcreteJob.PercentComplete = 0.
- 1548 2) ConcreteJob provider prompts client to respond with a DIAG35 message.
- 1549 3) ConcreteJob provider sets ConcreteJob.JobState = 12 (Query Pending) to wait for client
1550 response.
- 1551 4) When JobSettingData.InteractiveTimeout is exceeded, the ConcreteJob provider examines the
1552 value of JobSettingData.ClientRetriesMax.
- 1553 5) If the value of JobSettingData.ClientRetriesMax is zero, or if the number of retries has been
1554 exceeded, the ConcreteJob provider sets ConcreteJob.JobState = 10 (Exception) provided
1555 JobSettingData.TerminateOnTimeout=TRUE.
- 1556 6) If the value of JobSettingData.ClientRetriesMax is non-zero, steps 2, 3, 4, and 5 are repeated.
- 1557 NOTE If the value of JobSettingData.ClientRetriesMax is non-zero, the ConcreteJob provider monitors how many
1558 timeouts that have occurred.

1559 9.4 Silent mode operation

1560 An interactive test can be run as a non-interactive test; that is, the test does not prompt the client for
1561 responses. Instead, the ConcreteJob provider uses the CIM_JobSettingData (Default) values
1562 JobSettingData.DefaultInputNames and JobSettingData.DefaultInputValues to run the diagnostic test
1563 instead of sending a DIAG34 message to prompt the client.

1564 NOTE Silent mode works only if JobSettingData.DefaultInputNames and JobSettingData.DefaultInputValues have
1565 all the values needed to run the interactive test.

1566 An interactive test can only be run if the diagnostic test supports this capability; that is, if
1567 DiagnosticServiceJobCapabilities.SilentModeSupported = "TRUE". To run in silent mode, the client sets
1568 JobSettingData.RunInSilentMode = "TRUE" before invoking the DiagnosticTest.RunDiagnosticService()
1569 extrinsic method.

1570 9.4.1 Running an Interactive Test in Silent Mode

1571 The DiagnosticServiceJobCapabilities.SilentModeSupported is TRUE and the default JobSettingData has
1572 RunInSilentMode set to TRUE and non-empty arrays for DefaultInputNames and DefaultInputValues.

- 1573 1) Client calls DiagnosticTest.RunDiagnosticService() with the JobSettings parameter NULL and
1574 the method returns 0 (Success) and the object path of the ConcreteJob instance and starts the
1575 diagnostic test. ConcreteJob.JobState = 4 (Running) and ConcreteJob.PercentComplete = 0.
- 1576 2) When the test requires input, the ConcreteJob provider sees that the test is to run in Silent Mode
1577 (RunInSilentMode = TRUE) so the ConcreteJob gets the DefaultInputValues for the
1578 DefaultInputNames the test is looking for and supplies them to the test
- 1579 • No DIAG34 message is issued to the client
- 1580 3) ConcreteJob provider leaves ConcreteJob.JobState = 4 (Running) and the test continues to
1581 execute.
- 1582 4) If the test requests additional inputs, steps 2 and 3 are repeated.
- 1583 5) After the test is completed, the appropriate test completion status message is sent (indicating
1584 success or failure).

1585 9.4.2 Running Silent Mode with invalid default values

1586 The DiagnosticServiceJobCapabilities.SilentModeSupported is TRUE and the client supplies its own
1587 JobSettingData (in the JobSettings parameter) with RunInSilentMode set to TRUE and non-empty arrays

1588 for DefaultInputNames and DefaultInputValues, but the default values are invalid. (The
1589 DefaultInputNames are valid, but the DefaultInputValues are invalid.)

- 1590 1) Client calls DiagnosticTest.RunDiagnosticService() with the JobSettings parameter and the
1591 method returns 0 (Success) and the object path of the ConcreteJob instance and starts the
1592 diagnostic test. ConcreteJob.JobState = 4 (Running) and ConcreteJob.PercentComplete = 0.
- 1593 2) When the test requires input, the ConcreteJob provider sees that the test is to run in Silent Mode
1594 (RunInSilentMode = TRUE) so the ConcreteJob gets the DefaultInputValues for the
1595 DefaultInputNames the test is looking for and supplies them to the test
 - 1596 • No DIAG34 message is issued to the client
- 1597 3) ConcreteJob provider leaves ConcreteJob.JobState = 4 (Running) and the test continues to
1598 execute.
- 1599 4) The test cannot process the input values, so the job issues a DIAG39 message indicating that the
1600 Default Values were not used
- 1601 5) The job then gets the Default values in the Default JobSettingData and supplies these to the test
- 1602 6) If the test requests additional inputs, steps 2, 3, 4 and 5 are repeated.
- 1603 7) After the test is completed, the appropriate test completion status message is sent (indicating
1604 success or failure).

1605 NOTE If the DefaultInputValues are valid, but the DefaultInputNames are invalid, the process would be the same.
1606 The test will not attempt to use the invalid name until it has a need for the input. If it only accesses names that are
1607 valid, then it does not matter that one (or more) of the names are invalid.

1608 9.5 Finding diagnostic jobs

1609 9.5.1 Finding all diagnostic tests executed on a system

1610 A client can find all the diagnostic tests executed on a System as follows. Assume that the client starts at
1611 a known System on which the diagnostic tests are run.

- 1612 1) From the System instance, the client calls the Associators operation using HostedDependency as
1613 the association class argument, and ConcreteJob is the result class argument, which returns all
1614 instances of ConcreteJob.
- 1615 2) For each ConcreteJob instance returned, the client calls the Associators operation using
1616 OwningJobElement as the association class argument, DiagnosticTest as the result class
1617 argument, and OwningElement as the result role argument. The operation returns the associated
1618 DiagnosticTest instances.

1619 9.5.2 Finding all diagnostic tests executed against a ManagedElement

1620 A client can find all diagnostic tests executed against a ManagedElement as follows. Assume that the
1621 client starts at a known ManagedElement instance, which represents the device to be tested.

- 1622 1) From the ManagedElement instance, the client calls the Associators operation using
1623 AffectedJobElement as the association class argument, ConcreteJob as the result class
1624 argument, and AffectingElement as the result role argument.
- 1625 2) For each ConcreteJob instance returned, the client calls the Associators operation using
1626 OwningJobElement as the association class argument, DiagnosticTest as the result class
1627 argument, and OwningElement as the result role argument. The operation returns the associated
1628 DiagnosticTest instances.

1629 9.6 Configuring a diagnostic job

1630 To run a diagnostic test, the client invokes the `RunDiagnosticService()` extrinsic method of
1631 `DiagnosticTest`. The `JobSettings` parameter may be an empty string, `NULL` or a string representing an
1632 embedded instance of `JobSettingData`. When `JobSettings` is an empty string or `NULL`, the job runs using
1633 the default settings which may or may not have been published by the implementation.

1634 Note that the diagnostic default job settings are represented by a `JobSettingData (Default)` subclass that
1635 may have extensions. If the client is aware of the extensions, they may be modified as well. If the client is
1636 unaware, the default values should be used.

1637 9.6.1 Getting the default job settings

1638 The client can obtain the default job settings for a diagnostic test as follows. Assume that the client starts
1639 at a known `DiagnosticTest` instance.

- 1640 1) From the `DiagnosticTest` instance, the client calls the `Associators` operation using
1641 `ElementSettingData` as the association class argument, `JobSettingData` as the result class
1642 argument, and `SettingData` as the result role argument. The operation returns `JobSettingData`
1643 instances.
- 1644 2) For each `JobSettingData` instance, the client calls the `References` operation using
1645 `ElementSettingData` as the result class. The operation returns `ElementSettingData` instances.
- 1646 3) For each `ElementSettingData` instance, the client determines whether the value of the
1647 `ElementSettingData.IsDefault` property is 1 ("Is Default"). If so, the `JobSettingData` instance
1648 represents the default job settings.

1649 9.6.2 Creating the job settings

1650 A client may create their own job settings to pass as an argument to the
1651 `DiagnosticTest.RunDiagnosticService()` extrinsic method as follows. Assume that the client starts at a
1652 known `DiagnosticTest` instance.

- 1653 1) The client can attempt to discover the default job settings of the `DiagnosticTest` instance. The
1654 `GetDefaultJobSettings` use case (subclause 9.6.1) describes the necessary steps.
- 1655 2) If the client wants to not use the default job settings, the client can attempt to find the associated
1656 `DiagnosticServiceJobCapabilities` instance by calling the `Associators` operation using
1657 `ElementCapabilities` as the association class argument, `DiagnosticServiceJobCapabilities` as the
1658 result class, and `Capabilities` as the result role.
- 1659 3) If Step 2 returns an instance, the client calls the `CreateGoalSettings()` extrinsic method of the
1660 returned `DiagnosticServiceJobCapabilities` instance. This operation returns an instance of
1661 `JobSettingData` containing default values. The client can modify any property values as desired. If
1662 a range of values is permitted for a property, the client should use only those values indicated in
1663 the `DiagnosticServiceJobCapabilities` instance.
- 1664 4) If Step 2 does not return an instance because the implementation of
1665 `DiagnosticServiceJobCapabilities` is optional, the client should use the default `JobSettingData`.

1666 9.7 Execute and control a job for a diagnostic test

1667 The `DiagnosticServiceJobCapabilities.RequestedStatesSupported` property indicates the permitted
1668 values of the `RequestedState` input parameter for the `ConcreteJob.RequestStateChange()` extrinsic
1669 method. Since `DiagnosticServiceJobCapabilities` is an optional class, a client may not be able to examine
1670 an instance to determine which values of `RequestedState` to use. If a client invokes
1671 `ConcreteJob.RequestStateChange()` to change to an unsupported state, the extrinsic method shall return
1672 4097 ("Invalid State Transition").

1673 9.7.1 Suspend a job for a diagnostic test

1674 The client can suspend the execution of a test by invoking the `ConcreteJob.RequestStateChange()`
 1675 extrinsic method on the `ConcreteJob` instance that is returned from the
 1676 `DiagnosticTest.RunDiagnosticService()` extrinsic method. Assume that the client starts at a known
 1677 `ConcreteJob` instance and that a `DiagnosticServiceJobCapabilities` instance exists.

- 1678 1) From the `ConcreteJob` instance, the client calls the `Associators` operation using
 1679 `OwningJobElement` as the association class argument, `DiagnosticTest` as the result class
 1680 argument, and `OwningElement` as the result role. The associated `DiagnosticTest` instance is
 1681 returned.
- 1682 2) From the `DiagnosticTest` instance, the client calls the `Associators` operation using
 1683 `ElementCapabilities` as the association class argument, `DiagnosticServiceJobCapabilities` as the
 1684 result class argument, and `Capabilities` as the result role. The associated
 1685 `DiagnosticServiceJobCapabilities` instance is returned.
- 1686 3) The client examines the `DiagnosticServiceJobCapabilities.RequestedStatesSupported` property.
 1687 If it contains the value of 3 ("Suspend"), the `ConcreteJob` can be suspended.
- 1688 4) The client invokes the `ConcreteJob.RequestStateChange()` extrinsic method where input
 1689 parameter `RequestedState` has the value of 3 ("Suspend").
- 1690 5) After the transition is completed successfully, the `ConcreteJob.JobState` property shall have the
 1691 value of 5 ("Suspended") and `ConcreteJob.TimeOfLastStateChange` property shall be set to the
 1692 current time.

1693 9.7.2 Resume a job for a diagnostic test

1694 The client can resume the execution of a test that was previously suspended by invoking the
 1695 `ConcreteJob.RequestStateChange()` extrinsic method on the `ConcreteJob` instance that is returned from
 1696 the `DiagnosticTest.RunDiagnosticService()` extrinsic method. Assume that the client starts at a known
 1697 `ConcreteJob` instance and that a `DiagnosticServiceJobCapabilities` instance exists.

- 1698 1) From the `ConcreteJob` instance, the client calls the `Associators` operation using
 1699 `OwningJobElement` as the association class argument, `DiagnosticTest` as the result class
 1700 argument, and `OwningElement` as the result role. The associated `DiagnosticTest` instance is
 1701 returned.
- 1702 2) From the `DiagnosticTest` instance, the client calls the `Associators` operation using
 1703 `ElementCapabilities` as the association class argument, `DiagnosticServiceJobCapabilities` as the
 1704 result class argument, and `Capabilities` as the result role. The associated
 1705 `DiagnosticServiceJobCapabilities` instance is returned.
- 1706 3) The client examines the `DiagnosticServiceJobCapabilities.RequestedStatesSupported` property.
 1707 If it contains the value of 2 ("Start"), the `ConcreteJob` can be resumed.
- 1708 4) The client invokes the `ConcreteJob.RequestStateChange()` extrinsic method where input
 1709 parameter `RequestedState` has the value of 2 ("Start").
- 1710 5) After the transition is completed successfully, the `ConcreteJob.JobState` property shall have the
 1711 value of 4 ("Running") and `ConcreteJob.TimeOfLastStateChange` property shall be set to the
 1712 current time.

1713 NOTE The `JobState` property may transition from the value 3 ("Starting") before the final transition to the value of
 1714 4 ("Running").

1715 9.7.3 Terminate a job for a diagnostic test

1716 The client can cleanly terminate the execution of a test by invoking the
1717 `ConcreteJob.RequestStateChange()` extrinsic method on the `ConcreteJob` instance that is returned from
1718 the `DiagnosticTest.RunDiagnosticService()` extrinsic method. Assume that the client starts at a known
1719 `ConcreteJob` instance and that a `DiagnosticServiceJobCapabilities` instance exists.

- 1720 1) From the `ConcreteJob` instance, the client calls the `Associators` operation using
1721 `OwningJobElement` as the association class argument, `DiagnosticTest` as the result class
1722 argument, and `OwningElement` as the result role. The associated `DiagnosticTest` instance is
1723 returned.
- 1724 2) From the `DiagnosticTest` instance, the client calls the `Associators` operation using
1725 `ElementCapabilities` as the association class argument, `DiagnosticServiceJobCapabilities` as the
1726 result class argument, and `Capabilities` as the result role. The associated
1727 `DiagnosticServiceJobCapabilities` instance is returned.
- 1728 3) The client examines the `DiagnosticServiceJobCapabilities.RequestedStatesSupported` property.
1729 If it contains the value of 4 (“Terminate”), the `ConcreteJob` can be terminated.
- 1730 4) The client invokes the `ConcreteJob.RequestStateChange()` extrinsic method where input
1731 parameter `RequestedState` has the value of 4 (“Terminate”).
- 1732 5) After the transition is completed successfully, the `ConcreteJob.JobState` property shall have the
1733 value of 8 (“Terminated”) and `ConcreteJob.TimeOfLastStateChange` property shall be set to the
1734 current time.

1735 NOTE The `JobState` property may transition to 6 (“Shutting Down”) before the final transition to 8 (“Terminated”).

1736 9.7.4 Kill a job for a diagnostic test

1737 The client can immediately abort the execution of a test, with no attempt to perform a clean shutdown, by
1738 invoking the `ConcreteJob.RequestStateChange()` extrinsic method on the `ConcreteJob` instance that is
1739 returned from the `DiagnosticTest.RunDiagnosticService()` extrinsic method. Assume that the client starts
1740 at a known `ConcreteJob` instance and that a `DiagnosticServiceJobCapabilities` instance exists.

- 1741 1) From the `ConcreteJob` instance, the client calls the `Associators` operation using
1742 `OwningJobElement` as the association class argument, `DiagnosticTest` as the result class
1743 argument, and `OwningElement` as the result role. The associated `DiagnosticTest` instance is
1744 returned.
- 1745 2) From the `DiagnosticTest` instance, the client calls the `Associators` operation using
1746 `ElementCapabilities` as the association class argument, `DiagnosticServiceJobCapabilities` as the
1747 result class argument, and `Capabilities` as the result role. The associated
1748 `DiagnosticServiceJobCapabilities` instance is returned.
- 1749 3) The client examines the `DiagnosticServiceJobCapabilities.RequestedStatesSupported` property.
1750 If it contains the value of 5 (“Kill”), the `ConcreteJob` can be aborted.
- 1751 4) The client invokes the `ConcreteJob.RequestStateChange()` extrinsic method where input
1752 parameter `RequestedState` has the value of 5 (“Kill”).
- 1753 5) After the transition is completed successfully, the `ConcreteJob.JobState` property shall have the
1754 value of 9 (“Killed”) and `ConcreteJob.TimeOfLastStateChange` property shall be set to the current
1755 time.

1756 9.8 Delete a job for a diagnostic test

1757 When the optional DiagnosticServiceJobCapabilities instance exists and its DeleteJobSupported property
1758 has the value of TRUE, a client can control how and when the ConcreteJob instance associated to a
1759 specific diagnostic test execution is deleted. Assume that the client has verified that
1760 DiagnosticServiceJobCapabilities.DeleteJobSupported has the value of TRUE. Also assume that the
1761 client starts at a known DiagnosticTest instance. See Table 4 for further information.

1762 9.8.1 Client deletes a job for a diagnostic test

1763 To configure the ConcreteJob instance to be deleted by the client rather than by the provider

- 1764 1) The client creates an embedded instance of JobSettingData where its DeleteOnCompletion
1765 property has the value of FALSE.
- 1766 2) The client invokes the DiagnosticTest.RunDiagnosticService() extrinsic method where the
1767 JobSettings input parameter has the value of the embedded instance of JobSettingData created
1768 in the previous step.
- 1769 3) After the diagnostic test is completed or otherwise terminates, the ConcreteJob instance shall
1770 remain indefinitely until the client performs the DeleteInstance operation on it.

1771 NOTE This assumes that DiagnosticServiceJobCapabilities.CleanupInterval is NULL; otherwise, the provider
1772 deletes the ConcreteJob instance after the configured interval.

1773 9.8.2 Provider deletes a job

1774 To configure the ConcreteJob instance to be deleted by the provider rather than by the client

- 1775 1) The client creates an embedded instance of JobSettingData where its DeleteOnCompletion
1776 property has the value of TRUE.
- 1777 2) The client invokes the DiagnosticTest.RunDiagnosticService() extrinsic method where the
1778 JobSettings input parameter has the value of the embedded instance of JobSettingData created
1779 in the previous steps.
- 1780 3) The provider removes the ConcreteJob instance at the time indicated by the value of the
1781 TimeBeforeRemoval property.

1782 **EXPERIMENTAL**

1783 **10 CIM elements**

1784 Table 15 shows the instances of CIM elements for this profile. Instances of the CIM elements shall be
 1785 implemented as described in Table 15. Clause 7 (“Implementation”) and Clause 8 (“Methods”) may
 1786 impose additional requirements on these elements. See [DSP1103](#) for other mandatory elements that
 1787 must be implemented.

1788 **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 JobSettingData)	Mandatory	See 10.5.
CIM_FilterCollection	Optional	See 10.6.
CIM_HostedDependency	Mandatory	See 10.7.
CIM_IndicationFilter	Mandatory	See 10.8.
CIM_JobSettingData (Client)	Optional	See 10.9.
CIM_JobSettingData (Default)	Mandatory	See 10.10.
CIM_MemberOfCollection	Optional	See 10.11.
CIM_OwningCollectionElement	Optional	See 10.12.
CIM_OwningJobElement	Mandatory	See 10.13.
CIM_RegisteredProfile	Mandatory	See 10.14.
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) and DiagnosticTest.ResumeWithInput is supported. Query Language = "DMTF:CQL" Name = "DMTF:Diagnostic Job Control:DIAG9" See 7.9.1
SELECT * FROM CIM_AlertIndication WHERE OwningEntity="DMTF" and MessageID="DIAG12"	Optional	Query Language =: "DMTF:CQL" Name = "DMTF:Diagnostic Job Control:DIAG12" See 7.9.2
SELECT * FROM CIM_AlertIndication WHERE OwningEntity="DMTF" and MessageID="DIAG19"	Mandatory	Query Language - "DMTF:CQL" Name = "DMTF:Diagnostic Job Control:DIAG19" See 7.9.3

Element Name	Requirement	Description
SELECT * FROM CIM_AlertIndication WHERE OwningEntity="DMTF" and MessageID="DIAG20"	Mandatory	Query Language = "DMTF:CQL" Name = "DMTF:Diagnostic Job Control:DIAG20" See 7.9.4
SELECT * FROM CIM_AlertIndication WHERE OwningEntity="DMTF" and MessageID="DIAG21"	Optional	Query Language = "DMTF:CQL" Name = "DMTF:Diagnostic Job Control:DIAG21" See 7.9.5
SELECT * FROM CIM_AlertIndication WHERE OwningEntity="DMTF" and MessageID="DIAG34"	Conditional	This indication is mandatory if CIM_DiagnosticTest.Characteristics has the value of 3 (Is Interactive) and DiagnosticTest.ResumeWithInput is supported. Query Language = "DMTF:CQL" Name = "DMTF:Diagnostic Job Control:DIAG34" See 7.9.6
SELECT * FROM CIM_AlertIndication WHERE OwningEntity="DMTF" and MessageID="DIAG35"	Conditional	This indication is mandatory if CIM_DiagnosticTest.Characteristics has the value of 3 (Is Interactive) and DiagnosticTest.ResumeWithAction is supported. Query Language = "DMTF:CQL" Name = "DMTF:Diagnostic Job Control:DIAG35" See 7.9.7
SELECT * FROM CIM_AlertIndication WHERE OwningEntity="DMTF" and MessageID="DIAG36"	Optional	Query Language = "DMTF:CQL" Name = "DMTF:Diagnostic Job Control:DIAG36" See 7.9.8
SELECT * FROM CIM_AlertIndication WHERE OwningEntity="DMTF" and MessageID="DIAG37"	Optional	Query Language = "DMTF:CQL" Name = "DMTF:Diagnostic Job Control:DIAG37" See 7.9.9
SELECT * FROM CIM_AlertIndication WHERE OwningEntity="DMTF" and MessageID="DIAG38"	Optional	Query Language = "DMTF:CQL" Name = "DMTF:Diagnostic Job Control:DIAG38" See 7.9.10
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" Name = "DMTF:Diagnostic Job Control:DIAG39" See 7.9.11
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" Name = "DMTF:Diagnostic Job Control:DIAG40" See 7.9.12

Element Name	Requirement	Description
SELECT * FROM CIM_AlertIndication WHERE OwningEntity="DMTF" and MessageID="DIAG48"	Conditional	This indication is mandatory if CIM_DiagnosticTest.Characteristics has the value of 3 (Is Interactive). Query Language = "DMTF:CQL" Name = "DMTF:Diagnostic Job Control:DIAG48" See 7.9.13
SELECT * FROM CIM_AlertIndication WHERE OwningEntity="DMTF" and MessageID="DIAG49"	Conditional	This indication is mandatory if CIM_DiagnosticTest.Characteristics has the value of 3 (Is Interactive). Query Language = "DMTF:CQL" Name = "DMTF:Diagnostic Job Control:DIAG49" See 10.14

1789 10.1 CIM_AffectedJobElement

1790 Although defined in [DSP1103](#), the CIM_AffectedJobElement class is listed here because the
1791 AffectedElement reference is scoped down to CIM_DiagnosticTest. The constraints listed in Table 16 are
1792 in addition to those specified in [DSP1103](#). See [DSP1103](#) for other mandatory elements that must be
1793 implemented.

1794 **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 executing the test.
ElementEffects	Optional	See 7.1.1.
OtherElementEffectsDescriptions	Optional	If ElementEffects contains the value 1 (Other), this property is Mandatory.

1795 10.2 CIM_ConcreteJob

1796 Each successful invocation of the CIM_DiagnosticTest.RunDiagnosticService() extrinsic method returns
1797 a CIM_ConcreteJob instance. Each CIM_ConcreteJob instance represents a diagnostic test execution.
1798 This class specializes CIM_ConcreteJob as defined in [DSP1103](#). The constraints listed in Table 17 are in
1799 addition to those specified in [DSP1103](#). See [DSP1103](#) for other mandatory elements that must be
1800 implemented.

1801 **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.

Properties	Requirement	Notes
ElapsedTime	Mandatory	See 7.2.7.
RequestedState	Mandatory	
PercentComplete	Mandatory	See 7.2.8.
TimeOfLastStateChange	Mandatory	See 7.2.9
RequestStateChange()	Mandatory	See 8.2.1.
ResumeWithInput()	Conditional	See 8.2.2.
ResumeWithAction()	Conditional	See 8.2.3.

1802 **10.3 CIM_DiagnosticServiceJobCapabilities**

1803 The CIM_DiagnosticServiceJobCapabilities is used to provide information about on the capabilities of the
 1804 job that is used to run the diagnostic test. Table 18 contains the requirements for the elements of this
 1805 class.

1806 **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.

1807 **10.4 CIM_ElementCapabilities (Job)**

1808 CIM_ElementCapabilities represents an association between a test and its capabilities to use a job to run
 1809 the test. Table 19 contains the requirements for elements of this class.

1810 **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.

1811 **10.5 CIM_ElementSettingData (Default JobSettingData)**

1812 CIM_ElementSettingData represents an association between a CIM_DiagnosticTest and the
 1813 CIM_JobSettingData (Default) for the job used to run the test. The ManagedElement is responsible for
 1814 the creation of the job. Table 20 contains the requirements for elements of this class. If

1815 CIM_DiagnosticTest (or a subclass) has more than one type of test, a separate CIM_JobSettingData
 1816 (Default) may be defined for each. However, all instances of the same type of test shall reference the
 1817 same CIM_JobSettingData (Default) instance.

1818 **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 have the value TRUE.

1819 **10.6 CIM_FilterCollection (ProfileSpecificFilterCollection)**

1820 CIM_FilterCollection represents a ProfileSpecificFilterCollection as defined in [DSP1054](#). It defines the
 1821 collection of all the alert indications of the Diagnostic Job Control profile. Table 21 contains the
 1822 requirements for elements of this class.

1823 **Table 21 – Class: CIM_FilterCollection**

Properties	Requirement	Notes
InstanceID	Mandatory	Key: See DSP1054
CollectionName	Mandatory	The property shall have the value “DMTF:Diagnostic Job Control: ProfileSpecifiedAlertIndicationFilterCollection”

1824 **10.7 CIM_HostedDependency**

1825 CIM_HostedDependency represents an association between the system on which a test is run and the
 1826 CIM_ConcreteJob that is used to run the test. Table 22 contains the requirements for elements of this
 1827 class.

1828 **Table 22 – 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 executing the test.

1829 **10.8 CIM_IndicationFilter (StaticIndicationFilter)**

1830 CIM_IndicationFilter represents a StaticIndicationFilter as defined in [DSP1054](#). It defines the format of all
 1831 the alert indication filters in the Diagnostic Job Control profile. Table 23 contains the requirements for
 1832 elements of this class.

1833

Table 23 – Class: CIM_IndicationFilter (StaticIndicationFilter)

Properties	Requirement	Notes
Name	Mandatory	Key: See the Name values as identified in Table 15.
CreationClassName	Mandatory	Key: See DSP1054 .
SystemName	Mandatory	Key: See DSP1054 .
SystemCreationClassName	Mandatory	Key: See DSP1054 .
SourceNamespaces[]	Mandatory	See DSP1054 .
IndividualSubscriptionSupported	Mandatory	See DSP1054 .
Query	Mandatory	See the Query values as identified in Table 15.
QueryLanguage	Mandatory	See the QueryLanguage values as identified in Table 15.

1834 **10.9 CIM_JobSettingData (Client)**

1835 This CIM_JobSettingData definition represents the JobSettings parameter the client passes when it
 1836 invokes the RunDiagnosticService() extrinsic method of the DiagnosticTest. Table 24 contains the
 1837 requirements for elements of this class.

1838

Table 24 – 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 ".*")

1839 **10.10 CIM_JobSettingData (Default)**

1840 This CIM_JobSettingData definition represents the default settings for the job used to run the diagnostic
 1841 test. Table 25 contains the requirements for elements of this class. Each CIM_DiagnosticTest (or a
 1842 subclass) shall have one associated instance of this class.

1843

Table 25 – 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.

Properties	Requirement	Notes
DefaultInputValues	Conditional	See 7.5.5.
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 ".*")

1844 10.11 CIM_MemberOfCollection (ProfileSpecificMemberOfCollection)

1845 CIM_MemberOfCollection represents an association between the profile specific FilterCollection and the
 1846 CIM_IndicationFilters for the alert indications. Table 26 contains the requirements for elements of this
 1847 class.

1848 **Table 26 – Class: CIM_MemberOfCollection**

Properties	Requirement	Notes
Collection	Mandatory	Key. Value shall reference the profile specific FilterCollection instance representing a filter collection containing the alert indication filters.
Member	Mandatory	Key. Value shall reference an Alert IndicationFilter instance representing a contained alert indication filter.

1849 10.12 CIM_OwningCollectionElement

1850 CIM_OwningCollectionElement represents an association between the IndicationService that controls the
 1851 profile specific FilterCollection and the profile specific CIM_FilterCollection for the alert indication filters.
 1852 Table 27 contains the requirements for elements of this class.

1853 **Table 27 – Class: CIM_OwningCollectionElement**

Properties	Requirement	Notes
OwningElement	Mandatory	Key. See DSP1054
OwnedElement	Mandatory	Key. Value shall be a reference to the DMTF profile specific Alert Indication FilterCollection instance

1854 10.13 CIM_OwningJobElement

1855 Although defined in [DSP1103](#), the CIM_OwningJobElement class is listed here because the
 1856 OwningElement reference is scoped down to CIM_DiagnosticTest, which is a subclass of
 1857 CIM_ManagedElement. The constraints listed in Table 28 are in addition to those specified in [DSP1103](#).
 1858 See [DSP1103](#) for other mandatory properties of CIM_HostedService that must be implemented.

1859

Table 28 – 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 executing the test.

1860 **10.14 CIM_RegisteredProfile**

1861 The CIM_RegisteredProfile class is defined in [DSP1033](#). The constraints listed in Table 29 are in addition
 1862 to those specified in [DSP1033](#). See [DSP1033](#) for other mandatory properties of CIM_RegisteredProfile
 1863 that must be implemented.

1864

Table 29 – Class: CIM_RegisteredProfile

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

ANNEX A (informative)

Change log

1865
1866
1867
1868

Version	Date	Description
0.1	2010-12-05	Initial Version
1.0.0a	2012-09-11	Work In Progress version
1.0.0b	2012-10-27	<p>Changed the version to 1.0.0b</p> <p>3 Terms and definitions</p> <ul style="list-style-type: none"> • Modified the definition of Job • Added a definition of test <p>5 Synopsis</p> <ul style="list-style-type: none"> • Changed the version to 1.0.0b <p>7 Implementation</p> <ul style="list-style-type: none"> • Added a subclause "7.8 Diagnostics Job Control Profile Indications Support" • Added a subclause "7.9 Diagnostics Job Control Alert Indications and Standard Messages" <p>10 CIM elements</p> <ul style="list-style-type: none"> • Edited the Indications entries <ul style="list-style-type: none"> - changed the format of QueryLanguage entries to avoid confusion - Added a Name element to the Description - Replaced the text with a reference to the appropriate Implementation subclause. • Added a class table for CIM_FilterCollection • Added a class table for CIM_IndicationFilter • Added a class table for CIM_MemberOfCollection • Added a class table for CIM_OwningCollectionElement
	2012-11-03	<p>Changed "section" to "subclause"</p> <p>Changed "JobSetting" to "JobSettings"</p> <p>Edited Methods section</p> <p>Edited Requirements on Indications, particularly the Conditional statements</p>
	2012-12-17	Integrated comments from Peter and updated the figure
		NEED TO FIX THE TABLES
	2013-05-20	Work in Progress version

1869