



1
2
3
4

Document Identifier: DSP0223

Date: 2015-02-19

Version: 1.1.0

5 **Generic Operations**

6 **Supersedes: 1.0.0**

7 **Document Type: Specification**

8 **Document Class: Normative**

9 **Document Status: Published**

10 **Document Language: en-US**

11 Copyright notice

12 Copyright © 2007–2015 Distributed Management Task Force, Inc. (DMTF). All rights reserved.

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

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27 withdrawn or modified after publication, and shall be indemnified and held harmless by any party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

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

Contents

34	Foreword	6
35	Acknowledgements	6
36	Document conventions	6
37	Typographical conventions	6
38	Experimental material	6
39	1 Scope	7
40	2 Normative references	7
41	3 Terms and definitions	8
42	4 Symbols and abbreviated terms	10
43	5 Concepts	10
44	5.1 Interaction model for generic operations	10
45	5.2 Generic operations mappings	12
46	5.2.1 Overview	12
47	5.2.2 Recommendations	12
48	5.3 Conformance to generic operations	13
49	5.3.1 Conformance of entire WBEM protocols or APIs	13
50	5.3.2 Conformance of single WBEM operations or API calls	13
51	5.3.3 Requirement levels for operation parameters	14
52	5.4 Generic types	14
53	5.4.1 CIM data types	14
54	5.4.2 NamespacePath	14
55	5.4.3 InstancePath	15
56	5.4.4 ClassPath	15
57	5.4.5 QualifierTypePath	15
58	5.4.6 InstanceSpecification	15
59	5.4.7 ClassSpecification	16
60	5.4.8 QualifierType	17
61	5.4.9 InstanceSpecificationWithPath	17
62	5.4.10 ClassSpecificationWithPath	17
63	5.4.11 QualifierTypeWithPath	17
64	5.4.12 ClassName	17
65	5.4.13 PropertyName	17
66	5.4.14 MethodName	17
67	5.4.15 ParameterValue	18
68	5.4.16 ReturnValue	18
69	5.4.17 QueryString	18
70	5.4.18 QueryLanguage	18
71	5.4.19 EnumerationContext	18
72	5.4.20 ListenerDestination	18
73	5.5 Success and failure	18
74	5.6 Preconditions and postconditions	19
75	5.7 Generic error messages	19
76	5.8 Consistency model	20
77	5.8.1 Definition of ACID properties	20
78	5.8.2 Time consistency within instance representations	21
79	5.8.3 Staleness of information returned	21
80	5.8.4 Isolation between operations	21
81	5.8.5 Duplicate return of CIM objects or object paths	21
82	5.8.6 Time consistency between returned CIM objects	22
83	5.8.7 Order of returned CIM objects	22
84	5.8.8 Validity of returned object paths	22
85	5.8.9 Effects of deleting an instance	22

86	6	Generic operations	24
87	6.1	Description format.....	25
88	6.2	Common operation parameters for all operations	27
89	6.2.1	IncludeClassOrigin (partly deprecated)	27
90	6.2.2	IncludeQualifiers	27
91	6.2.3	<element>List	28
92	6.3	Instance operations.....	28
93	6.3.1	GetInstance.....	28
94	6.3.2	DeleteInstance.....	30
95	6.3.3	ModifyInstance.....	32
96	6.3.4	CreateInstance.....	34
97	6.4	Direct instance enumeration operations	37
98	6.4.1	EnumerateInstances (deprecated)	37
99	6.4.2	EnumerateInstanceNames (deprecated).....	40
100	6.4.3	Associators (deprecated).....	41
101	6.4.4	AssociatorNames (deprecated)	45
102	6.4.5	References (deprecated)	48
103	6.4.6	ReferenceNames (deprecated)	51
104	6.5	Pulled instance enumeration operations.....	53
105	6.5.1	General behavioral rules.....	54
106	6.5.2	Common operation parameters for the open operations	56
107	6.5.3	OpenEnumerateInstances	58
108	6.5.4	OpenEnumerateInstancePaths (deprecated)	62
109	6.5.5	OpenAssociators.....	65
110	6.5.6	OpenAssociatorPaths (deprecated).....	70
111	6.5.7	OpenReferences.....	75
112	6.5.8	OpenReferencePaths (deprecated).....	80
113	6.5.9	OpenQueryInstances	84
114	6.5.10	Common operation parameters for the pull operations	86
115	6.5.11	PullInstancesWithPath	87
116	6.5.12	PullInstancePaths (deprecated).....	89
117	6.5.13	PullInstances.....	92
118	6.5.14	CloseEnumeration	94
119	6.5.15	EnumerationCount (deprecated)	95
120	6.6	Method invocation operations	97
121	6.6.1	InvokeMethod	97
122	6.6.2	InvokeStaticMethod	99
123	6.7	Class operations	101
124	6.7.1	GetClass	101
125	6.7.2	DeleteClass.....	104
126	6.7.3	ModifyClass	107
127	6.7.4	CreateClass	109
128	6.8	Class enumeration operations	111
129	6.8.1	EnumerateClasses.....	111
130	6.8.2	EnumerateClassNames	114
131	6.8.3	AssociatorClasses	116
132	6.8.4	AssociatorClassPaths	119
133	6.8.5	ReferenceClasses.....	121
134	6.8.6	ReferenceClassPaths	124
135	6.9	Qualifier type operations	126
136	6.9.1	GetQualifierType.....	126
137	6.9.2	DeleteQualifierType	127
138	6.9.3	ModifyQualifierType	129
139	6.9.4	CreateQualifierType.....	130
140	6.9.5	EnumerateQualifierTypes	132
141	6.10	Indication delivery operations	133

142	6.10.1 DeliverIndication	134
143	ANNEX A (informative) Future operations	136
144	A.1 Test for property modifiability	136
145	A.2 Retrieval of associated instance graph	136
146	ANNEX B (informative) Changed generic operation names	137
147	ANNEX C (normative) Cross-namespace associations	139
148	C.1 Binary association using same schema version	139
149	ANNEX D (informative) Change log	143
150	Bibliography	145
151		

152 Figures

153	Figure 1 – Interaction model for generic server operations	11
154	Figure 2 – Interaction model for generic listener operations	11
155	Figure 3 – Generic operations mappings	12
156	Figure C-1 – Typical profile representation of binary association crossing namespaces	139
157	Figure C-2 – Binary association: WBEM server objects for bidirectional traversal	140
158	Figure C-3 – Binary association: WBEM server objects for unidirectional traversal	142
159		

160 Tables

161	Table 1 – List of generic operations	24
162	Table B-1 – Changed generic operation names	137
163		

164

Foreword

165 The *Generic Operations* specification (DSP0223) was originally prepared by the Generic Operations
166 Working Group of the DMTF and is now owned by the Architecture Working Group of the DMTF.

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

169 Acknowledgements

170 DMTF acknowledges the following individuals for their contributions to this specification:

- 171 • Jim Davis, WBEM Solutions
- 172 • George Ericson, EMC
- 173 • Steve Hand, Symantec
- 174 • Jon Hass, Dell
- 175 • Lawrence Lamers, VMware
- 176 • Andreas Maier, IBM (editor)
- 177 • Karl Schopmeyer, Inova Development

178 Document conventions

179 Typographical conventions

180 The following typographical conventions are used in this document:

- 181 • The titles of referenced documents are marked in *italics*.
- 182 • Important terms that are used for the first time are marked in *italics*.
- 183 • Generic parameters and generic types are marked in *italics*.
- 184 • The usage of terms typically links to their definition. Example: class path
- 185 • XML text is in `monospaced font`.

186 Experimental material

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

192 The following typographical convention indicates experimental material:

193 **EXPERIMENTAL**

194 Experimental material appears here.

195 **EXPERIMENTAL**

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

198

Generic Operations

199 1 Scope

200 DMTF defines a number of protocols that describe how managed resources that are modeled using CIM
201 can be discovered, accessed and manipulated:

- 202 • CIM-XML: The protocol defined in the *CIM Operations over HTTP Specification* ([DSP0200](#)), the
203 *Representation of CIM in XML Specification* ([DSP0201](#)) and the *DTD for Representation of CIM*
204 *in XML* ([DSP0203](#)).
- 205 • WS-Management: The usage of the WS-Management protocol for CIM, as defined in the *WS-*
206 *Management CIM Binding Specification* ([DSP0227](#)), the *WS-CIM Mapping Specification*
207 ([DSP0230](#)), the *Web Services for Management Specification* ([DSP0226](#)), and other underlying
208 Web Services specifications.
- 209 • CIM-RS: The RESTful protocol for CIM, as defined in *CIM-RS Protocol* ([DSP0210](#)) and in *CIM-*
210 *RS Payload Representation in JSON* ([DSP0211](#)).
- 211 • SM-CLP: The protocol defined in the *Server Management Command Line Protocol Specification*
212 ([DSP0214](#)), covering the core of the protocol common for all management profiles, and SM-
213 CLP mapping specifications for each management profile, covering profile-specific aspects of
214 the protocol such as verbs for extrinsic methods.

215 As different as these protocols are, they have certain operations and semantics in common, at least when
216 looking at it from a higher level. These common semantics can be used to define generic operations. This
217 specification defines an operational model and behavior associated to these operations at an abstracted,
218 generic level, and common across these protocols.

219 The generic operations are expected to be used in the following areas:

- 220 • Future releases of management profiles can define requirements on intrinsic operations by
221 referencing generic operations. Currently, they do that by referencing the operations defined for
222 the CIM-XML protocol. Using generic operations allows management profiles to become
223 independent of protocols. Management profiles defined in XML using the *Management Profile*
224 *XML Schema* ([DSP8028](#)) are required to use generic operations.
- 225 • Future and existing DMTF protocols can define mappings between their protocol-specific
226 operations and the generic operations. This drives more commonality across these protocols,
227 and consequently makes it easier to support multiple protocols in client applications, server side
228 instrumentation, and mapping bridges between protocols (also known as protocol gateways).
- 229 • Client APIs, server APIs and provider APIs can define their API calls conformant to the generic
230 operations. This drives more commonality across these APIs and between these APIs and
231 WBEM protocols, and consequently makes it easier to support multiple protocols with the same
232 API in client libraries and server side instrumentation (e.g., provider APIs).

233 2 Normative references

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

238 DMTF DSP0004, *CIM Infrastructure Specification 2.8*,
239 http://www.dmtf.org/standards/published_documents/DSP0004_2.8.pdf

- 240 DMTF DSP0198, *WBEM Glossary 1.0*,
241 http://www.dmtf.org/standards/published_documents/DSP0198_1.0.pdf
- 242 DMTF DSP0207, *WBEM URI Mapping 1.0*,
243 http://www.dmtf.org/standards/published_documents/DSP0207_1.0.pdf
- 244 DMTF DSP0212, *Filter Query Language 1.0.1*,
245 http://www.dmtf.org/standards/published_documents/DSP0212_1.0.1.pdf
- 246 DMTF DSP1054, *Indications Profile 1.2*,
247 http://www.dmtf.org/standards/published_documents/DSP1054_1.2.pdf
- 248 DMTF DSP8016, *WBEM Operations Message Registry 1.1*,
249 http://schemas.dmtf.org/wbem/messageregistry/1/dsp8016_1.1.xml
- 250 ISO/IEC Directives, Part 2:2004, *Rules for the structure and drafting of International Standards*,
251 <http://isotc.iso.org/livelink/livelink?func=ll&objId=4230456&objAction=browse>

252 3 Terms and definitions

253 In this specification, some terms have a specific meaning beyond the normal English meaning. Those
254 terms are defined in this clause.

255 The terms "shall" ("required"), "shall not", "should" ("recommended"), "should not" ("not recommended"),
256 "may", "need not" ("not required"), "can" and "cannot" in this specification are to be interpreted as
257 described in [ISO/IEC Directives, Part 2](#), Annex H. The terms in parenthesis are alternatives for the
258 preceding term, for use in exceptional cases when the preceding term cannot be used for linguistic
259 reasons. [ISO/IEC Directives, Part 2](#), Annex H specifies additional alternatives. Occurrences of such
260 additional alternatives shall be interpreted in their normal English meaning.

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

263 The terms "normative" and "informative" in this specification are to be interpreted as described in [ISO/IEC](#)
264 [Directives, Part 2](#), Clause 3. In this specification, clauses, subclauses or annexes indicated with
265 "(informative)" as well as notes and examples do not contain normative content.

266 The terms "class path", "creation class", "instance path", "management profile", "namespace path",
267 "object", "object path", "qualifier type path", "WBEM client", "client", "WBEM listener", "listener", "WBEM
268 server", "server", "WBEM operation", "WBEM protocol", and any other terms defined in [DSP0198](#) apply to
269 this specification. The following additional terms are used in this document.

270 3.1

271 duplicate object

272 objects in a result set that have duplicate object paths.

273 3.2

274 duplicate object path

275 object paths in a result set that reference the same object accessible through the WBEM server.

276 3.3

277 effective qualifier value

278 The effective value of a qualifier specified on a schema element is the value that determines the qualifier
279 behavior for the schema element, taking the qualifier propagation rules into account. For a complete
280 definition, see [DSP0004](#).

- 281 **3.4**
282 **exposed elements of a class**
283 The set of schema elements exposed by a class (i.e., properties and methods) is the union of the set of
284 elements defined in the class (including overridden elements) and the set of inherited elements that are
285 not overridden in the class. For a complete definition, see [DSP0004](#).
- 286 **3.5**
287 **generic listener operation**
288 a generic operation directed from a WBEM server to a WBEM listener. Also called listener operation. For
289 details, see 5.1.
- 290 **3.6**
291 **generic operation**
292 a generic operation as defined in this specification. Also called operation. They are divided into generic
293 listener operations and generic server operations. For details, see 5.1.
- 294 **3.7**
295 **generic operation request**
296 the request portion of a generic operation. Also called operation request. For details, see 5.1.
- 297 **3.8**
298 **generic operation response**
299 the response portion of a generic operation. Also called operation response. For details, see 5.1.
- 300 **3.9**
301 **generic operations mapping**
302 a mapping of generic operations to the operations of some other protocol (e.g., WBEM operations) or to
303 the calls of some API, as defined in 5.2.
- 304 **3.10**
305 **generic server operation**
306 a generic operation directed from a WBEM client to a WBEM server. Also called server operation. For
307 details, see 5.1.
- 308 **3.11**
309 **isolation**
310 the set of behaviors that describe how the execution of an operation affects the execution of another,
311 concurrent operation, as defined in 5.8.4.
- 312 **3.12**
313 **volatile property**
314 a property in an instance whose value may change as a WBEM client obtains the instance repeatedly
315 without performing any client-originated updates to the property value.
- 316 **3.13**
317 **WBEM listener operation**
318 a WBEM operation that is originated on a WBEM server and processed by a WBEM listener. For details,
319 see 5.1.
- 320 **3.14**
321 **WBEM protocol mapping**
322 a mapping of generic operations to a WBEM protocol, as defined in 5.2.

323 **3.15**

324 **WBEM server operation**

325 a WBEM operation that is originated by a WBEM client and processed by a WBEM server. For details,
326 see 5.1.

327 **4 Symbols and abbreviated terms**

328 The abbreviations "API", "CIM", "CIM-XML", "CIM-RS", "CQL", "UML", "WBEM", "WS-Management",
329 "XML", and any other symbols and abbreviations defined in [DSP0198](#) apply to this specification. The
330 following additional abbreviations are used in this document.

331 **4.1**

332 **SM-CLP**

333 Server Management Command Line Protocol, defined in [DSP0214](#)

334 **5 Concepts**

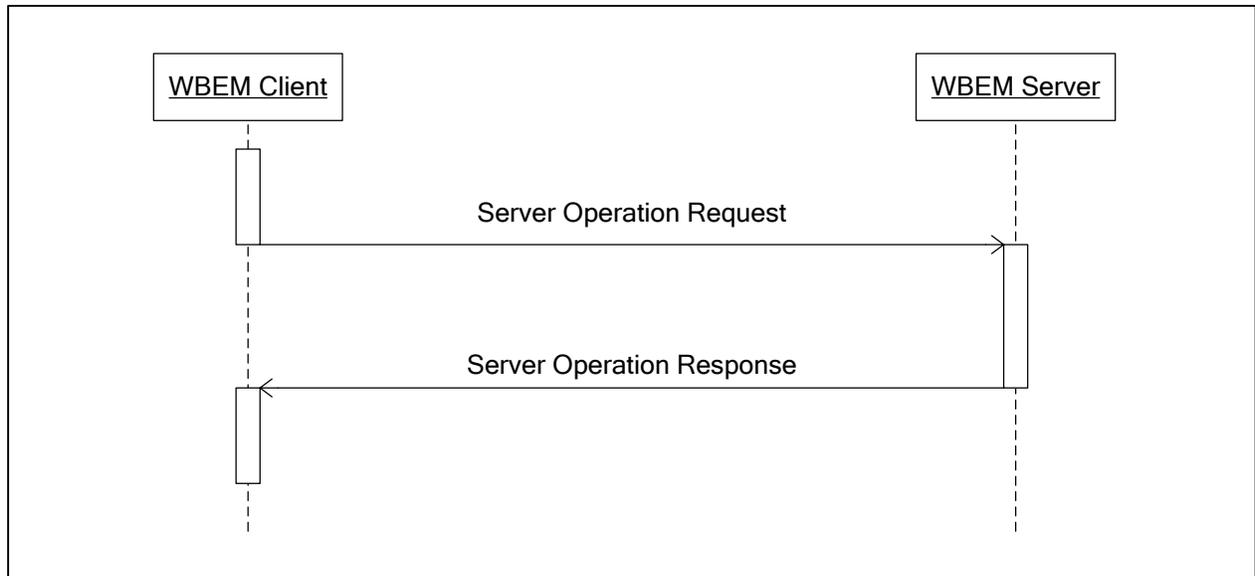
335 This clause defines concepts that are the basis for the definition of the generic operations.

336 **5.1 Interaction model for generic operations**

337 Generic operations are divided into two categories:

- 338 • **Generic server operations:** An operation request is sent from a WBEM client to a WBEM
339 server in order to initiate the processing of the operation, and an operation response is sent
340 back from the server to the client upon completion of the operation.
- 341 • **Generic listener operations:** An operation request is sent from a WBEM server to a WBEM
342 listener in order to initiate the processing of the operation, and an operation response is sent
343 back from the listener to the server upon completion of the operation.

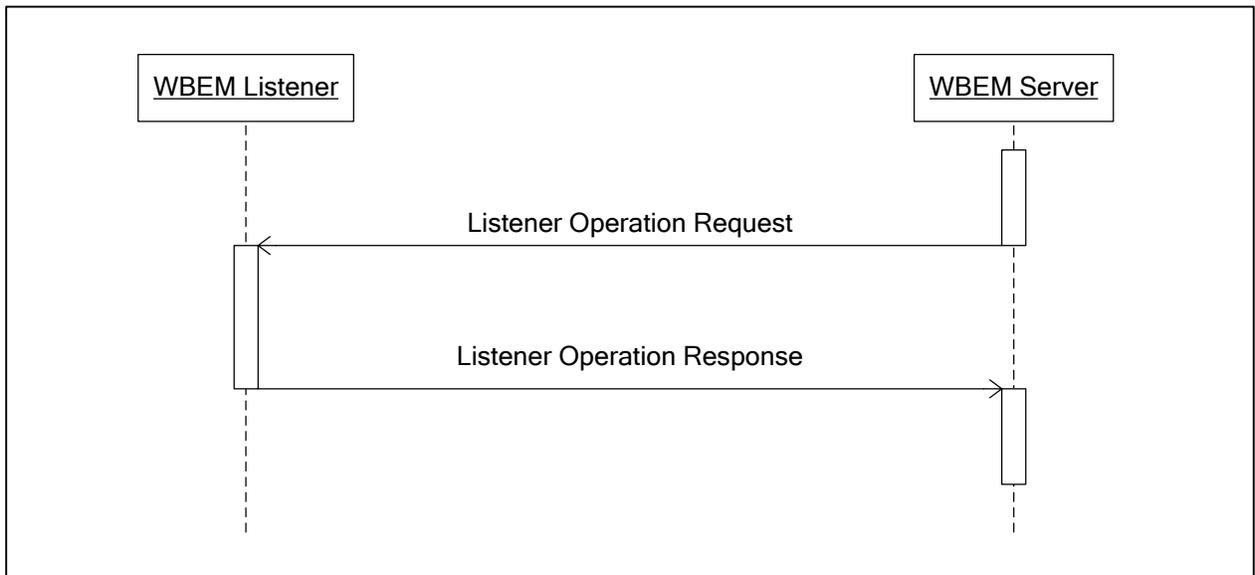
344 Figure 1 shows the interaction model for generic server operations, using a UML sequence diagram:



345
346

Figure 1 – Interaction model for generic server operations

347
348 Figure 2 shows the interaction model for generic listener operations, using a UML sequence diagram:



349
350

Figure 2 – Interaction model for generic listener operations

352 The operation request and operation response at the level of generic operations do not necessarily need
 353 to correspond directly to WBEM operations, that is to messages that are flowing at the level of the WBEM
 354 protocol. For example, a generic operation response may be delivered asynchronously at the level of the
 355 WBEM protocol.

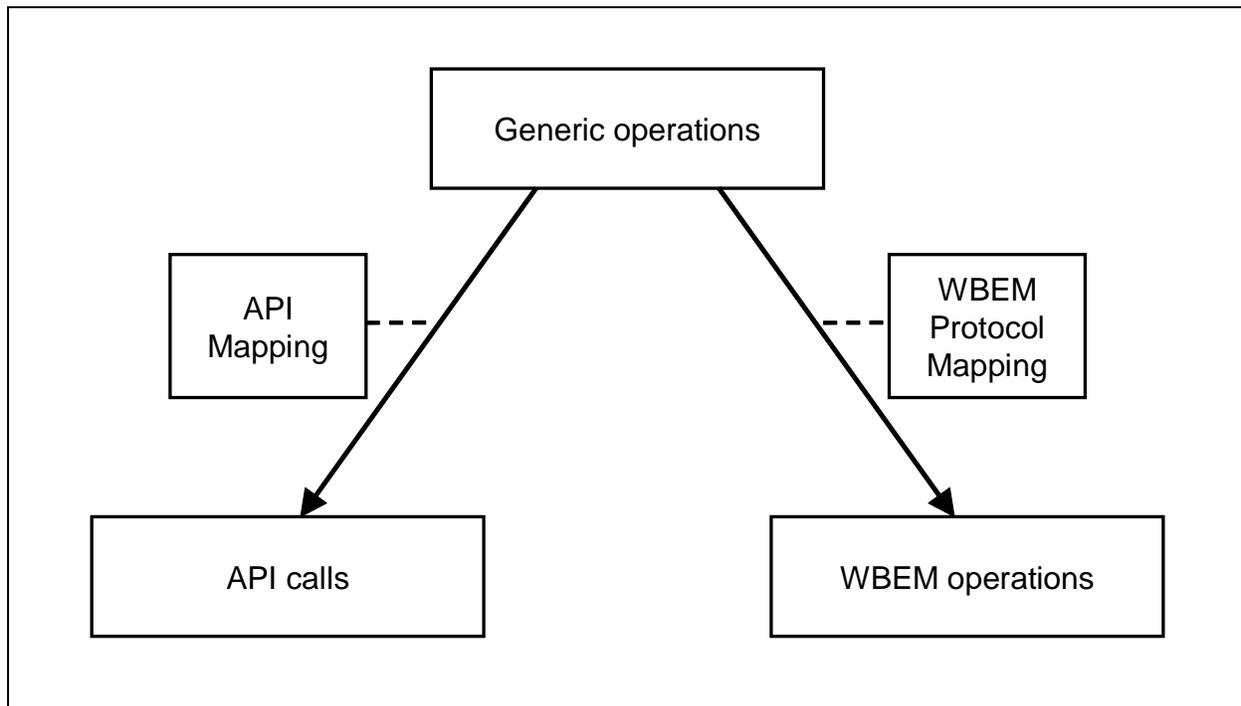
356 At the level of generic operations, any *input parameters* are part of the operation request, and any *output*
 357 *parameters* are part of the operation response. A WBEM protocol may choose to do that differently, for
 358 example by pushing some of the input parameters to the server in the form of options that are set, and
 359 that are used during the processing of subsequent operations.

360 This abstraction of generic operations from WBEM operations allows keeping the definition of the generic
 361 operations simple and scoped to defining the operation semantics. The details about the actual message
 362 flows are left to the scope of WBEM protocols. This separation is key in order to use the same definition
 363 of generic operations for multiple WBEM protocols.

364 **5.2 Generic operations mappings**

365 **5.2.1 Overview**

366 Figure 3 shows mappings of generic operations to WBEM protocols and APIs. These mappings allow
 367 determining which WBEM operations or API calls need to be implemented for a particular generic
 368 operation to be supported. This is used for example when implementing management profiles that specify
 369 provisions for intrinsic operations by referencing generic operations.



370

371 **Figure 3 – Generic operations mappings**

372 **5.2.2 Recommendations**

373 This subclause provides recommendations for specifying WBEM protocol mappings and API mappings
 374 that provide for determining the WBEM operations or API calls that support a particular generic operation,
 375 and specify conformance.

376 There is no requirement that WBEM protocol mappings and API mappings are defined in a separate
 377 specification (i.e., they can be defined in the specifications that define the WBEM protocol or API).

378 The following recommendations apply:

- 379 • WBEM protocol mappings and API mappings should define the mapping from a perspective of
380 the generic operation (i.e., by listing the relevant generic operation at the top level).
- 381 • For each generic operation listed in the mapping, the corresponding WBEM operations or API
382 calls should be stated that provide the functionality supporting the generic operation.
- 383 • For each parameter defined for a generic operation listed in the mapping, the corresponding
384 parameters and return values of the WBEM operations or API calls should be stated.
- 385 • A statement should be made for each generic operation as to whether or not the operation is
386 supported in a conformant way, as defined in 5.3.2. If the operation is supported in a non-
387 conformant way, the deviations should be stated.
- 388 • A statement should be made for the entire WBEM protocol or API as to whether or not it is
389 conformant to generic operations.

390 5.3 Conformance to generic operations

391 Conformance of a WBEM protocol or API to generic operations is defined at two levels:

- 392 1) At the level of the entire WBEM protocol or API
- 393 2) At the level of single WBEM operations or single API calls

394 The guiding principle for conformance to generic operations is that a WBEM protocol or API call is able to
395 completely represent the generic operations and their semantics. Functionalities of the WBEM protocol or
396 API that go beyond the functionality of generic operations are not relevant for conformance.

397 5.3.1 Conformance of entire WBEM protocols or APIs

398 A WBEM protocol or API is conformant to generic operations if all generic operations defined in this
399 specification are supported by WBEM operations or API calls in a conformant way, as defined in 5.3.2.

400 Conformant WBEM protocols or APIs may define WBEM operations or API calls in addition to those that
401 are mapped to generic operations.

402 5.3.2 Conformance of single WBEM operations or API calls

403 A particular generic operation is supported by WBEM operations or API calls in a conformant way if all of
404 the following is satisfied:

- 405 • The generic operation has one or more corresponding WBEM operations or API calls that
406 provide the functionality of the generic operation. The names of these corresponding WBEM
407 operations or API calls may be different from the name of the generic operation.
- 408 • Functionalities that are required to be supported for a generic operation are supported by the
409 corresponding WBEM operations or API calls with the semantics defined by the generic
410 operation.
- 411 • If functionalities that are optional to be supported for a generic operation are supported by the
412 corresponding WBEM operations or API calls, they are supported with the semantics defined by
413 the generic operation.
- 414 • Each parameter of a generic operation is mapped to one or more corresponding parameters of
415 the corresponding WBEM operations or API calls
- 416 • For each parameter of a generic operation, the provisions defined in 5.3.3 are satisfied.

417 WBEM operations or API calls that support a generic operation in a conformant way, may support
418 parameters or return values in addition to the parameters mapped to parameters of the corresponding

419 generic operation. Defining additional parameters can affect the ability to transform one WBEM protocol
420 into another (e.g., in protocol gateways).

421 5.3.3 Requirement levels for operation parameters

422 The parameters defined for generic operations each have a requirement level, as defined in this
423 subclause. That requirement level defines whether a conformant WBEM protocol or API has to support
424 the parameter.

425 The allowable requirement levels for parameters of generic operations are:

426 **Mandatory**

427 Operation parameters designated as mandatory shall be supported by conformant WBEM
428 protocols or APIs with the semantics defined for the generic operation. Conformant WBEM
429 protocols or APIs may define that supplying values for the corresponding parameters is optional
430 if a default behavior is specified.

431 **Conditional**

432 Operation parameters designated as conditional shall be supported by conformant WBEM
433 protocols or APIs if the specified condition is met. If supported, they shall be supported as
434 defined for the generic operation. Conformant WBEM protocols or APIs may define that
435 supplying values for the corresponding parameters is optional if a default behavior is specified.

436 **Optional**

437 Operation parameters designated as optional may be supported by conformant WBEM
438 protocols or APIs. If supported, they shall be supported as defined for the generic operation.
439 Conformant WBEM protocols or APIs may define that supplying values for the corresponding
440 parameters is optional if a default behavior is specified.

441 NOTE Conformant WBEM protocols or APIs may specify that supplying values for a supported parameter is
442 optional as long as the protocol or API defines a default value for the parameter. In other words, there are two
443 different kinds of requirements related to parameters:

- 444 1. The requirement to support a parameter in a WBEM protocol or API as defined by its requirement level
- 445 2. The requirement defined by the WBEM protocol or API for supplying a value for a supported parameter when
446 invoking an operation

447 5.4 Generic types

448 This specification defines the following generic data types for use by operation parameters of generic
449 operations.

450 5.4.1 CIM data types

451 All CIM data types defined in [DSP0004](#) (e.g., boolean) may be used as generic types. Values of these
452 data types can assume the (untyped) value NULL, as defined in [DSP0004](#).

453 5.4.2 NamespacePath

454 A value of the generic type *NamespacePath* represents a namespace path as defined in [DSP0004](#).

455 This specification does not define particular sub-components of a namespace path; as a result, any
456 requirements on the presence of such sub-components are left to conformant WBEM protocols.

457 Conformant WBEM protocols shall support all characteristics of *NamespacePath* values and may support
458 additional characteristics.

459 5.4.3 InstancePath

460 A value of the generic type *InstancePath* represents an instance path as defined in [DSP0004](#).

461 An *InstancePath* value shall specify the class name and key binding components of the represented
462 instance path. Any requirements for specifying or omitting the namespace path component in an
463 *InstancePath* value are left to conformant WBEM protocols.

464 Conformant WBEM protocols shall support all characteristics of *InstancePath* values and may support
465 additional characteristics.

466 5.4.4 ClassPath

467 A value of the generic type *ClassPath* represents a class path as defined in [DSP0004](#).

468 A *ClassPath* value shall specify the class name component of the represented class path. Any
469 requirements for specifying or omitting the namespace path component in a *ClassPath* value are left to
470 conformant WBEM protocols.

471 Conformant WBEM protocols shall support all characteristics of *ClassPath* values and may support
472 additional characteristics.

473 5.4.5 QualifierTypePath

474 A value of the generic type *QualifierTypePath* represents a qualifier type path as defined in [DSP0004](#).

475 A *QualifierTypePath* value shall specify the qualifier name component of the represented qualifier type
476 path. Any requirements for specifying or omitting the namespace path component in a *QualifierTypePath*
477 value are left to conformant WBEM protocols.

478 Conformant WBEM protocols shall support all characteristics of *ClassPath* values may support additional
479 characteristics.

480 5.4.6 InstanceSpecification

481 A value of the generic type *InstanceSpecification* is a representation of a CIM instance as defined for the
482 *Instance* meta-element defined in [DSP0004](#), containing:

- 483 • name of the creation class of the instance
- 484 • all or a subset of the static and non-static properties exposed by the creation class of the
485 instance

486 Each property in an *InstanceSpecification* shall contain:

- 487 • name of the property
- 488 • value of the property
- 489 • optional: Class origin of the property
- 490 • optional: Data type of the property

491 *InstanceSpecification* does not contain the instance path of the instance, because there are some
492 situations in which the instance data is needed without an instance path. The
493 *InstanceSpecificationWithPath* type is used when the instance path is needed in addition to the instance
494 data.

495 Generic operations using this type define the rules for the optional items in the content of this type.

496 5.4.7 ClassSpecification

497 A value of the generic type *ClassSpecification* is a representation of a CIM class as defined for the *Class*
498 meta-element defined in [DSP0004](#), containing:

- 499 • name of the class
- 500 • name of the superclass, if any
- 501 • all or a subset of the static and non-static properties (that is, the property definitions) exposed
502 by the class. As defined in DSP0004, the set of properties exposed by a class includes any
503 properties inherited from superclasses, where overridden properties are included only once.
- 504 • all of the static and non-static methods exposed by the class. As defined in DSP0004, the set of
505 methods exposed by a class includes any methods inherited from superclasses, where
506 overridden methods are included only once.
- 507 • optional: all of the qualifiers exposed by the class that are defined on the class or any of its
508 superclasses

509 Each property in a *ClassSpecification* shall contain:

- 510 • name of the property
- 511 • data type of the property
- 512 • default value of the property
- 513 • optional: all of the qualifiers exposed by the property that are defined on the property or any of
514 its overridden properties

515 Each method in a *ClassSpecification* shall contain:

- 516 • name of the method
- 517 • data type of the return value of the method
- 518 • all of the parameters of the method
- 519 • optional: all of the qualifiers exposed by the method that are defined on the method or any of its
520 overridden methods

521 Each parameter in that method shall contain:

- 522 • name of the parameter
- 523 • data type of the parameter
- 524 • optional: all of the qualifiers exposed by the parameter that are defined on the parameter or the
525 corresponding parameter in any of its overridden methods

526 Each qualifier in any of the items above shall contain:

- 527 • name of the qualifier
- 528 • effective value of the qualifier, as seen in the scope of the class represented by *Class*

529 *ClassSpecification* does not contain the class path of the class. The *ClassSpecificationWithPath* type is
530 used when the class path is needed in addition to the class.

531 Generic operations using this type define the rules for the optional items in the content of this type.

532 5.4.8 QualifierType

533 A value of the generic type *QualifierType* is a representation of a CIM qualifier type as defined for the
534 *QualifierType* meta-element defined in [DSP0004](#) (i.e., a qualifier declaration) containing:

- 535 • name of the qualifier
- 536 • data type of the qualifier
- 537 • default value of the qualifier
- 538 • all flavors of the qualifier
- 539 • all scopes of the qualifier

540 *QualifierType* does not contain the qualifier type path of the qualifier type. The *QualifierTypeWithPath*
541 type is used when the qualifier type path is needed in addition to the qualifier type.

542 5.4.9 InstanceSpecificationWithPath

543 A value of the generic type *InstanceSpecificationWithPath* combines the content of *InstanceSpecification*
544 and *InstancePath*.

545 *InstanceSpecification* shall represent the instance referenced by *InstancePath*.

546 5.4.10 ClassSpecificationWithPath

547 A value of the generic type *ClassSpecificationWithPath* combines the content of *ClassSpecification* and
548 *ClassPath*.

549 *ClassSpecification* shall represent the class referenced by *ClassPath*.

550 5.4.11 QualifierTypeWithPath

551 A value of the generic type *QualifierTypeWithPath* combines the content of *QualifierType* and
552 *QualifierTypePath*.

553 *QualifierType* shall represent the qualifier type referenced by *QualifierTypePath*.

554 5.4.12 ClassName

555 A value of the generic type *ClassName* is the name of a CIM class, including its schema prefix.

556 5.4.13 PropertyName

557 A value of the generic type *PropertyName* is the name of a CIM property or reference.

558 The class defining the property is not identified by the data in this type.

559 5.4.14 MethodName

560 A value of the generic type *MethodName* is the name of a CIM method.

561 The class defining the method and the method signature are not identified by the data in this type.

562 5.4.15 ParameterValue

563 A value of the generic type *ParameterValue* is a parameter value used as an input or output parameter
564 during invocation of a CIM method, containing:

- 565 • name of the parameter
- 566 • value of the parameter
- 567 • optional: Data type of the parameter

568 Generic operations using this type define the rules for the optional items in the content of this type.

569 5.4.16 ReturnValue

570 A value of the generic type *ReturnValue* is the value returned by the invocation of a CIM method,
571 containing:

- 572 • return value
- 573 • optional: Data type of the return value

574 Generic operations using this type define the rules for the optional items in the content of this type.

575 5.4.17 QueryString

576 A value of the generic type *QueryString* is a query string in some query language. The query language is
577 not identified by the data in this type.

578 5.4.18 QueryLanguage

579 A value of the generic type *QueryLanguage* is a query language of a query string.

580 5.4.19 EnumerationContext

581 A value of the generic type *EnumerationContext* is a value that uniquely identifies an enumeration
582 session used in pulled instance enumeration operations. It is opaque to WBEM clients.

583 5.4.20 ListenerDestination

584 A value of the generic type *ListenerDestination* is a value that uniquely addresses a WBEM listener for
585 purposes of delivering an indication to it using the *DeliverIndication* operation (see 6.10.1).

586 The format of the address is defined by the WBEM protocol.

587 5.5 Success and failure

588 All generic operations either succeed or fail. There is no concept of "partial success".

589 If a generic operation succeeds, it delivers its output data back to the operation requester, and does not
590 include any error messages.

591 If it fails, it delivers back one or more error messages, and no output data. For details about error
592 messages, see 5.7.

593 For example, if an instance enumeration operation were able to return some instances successfully, but
594 not all successfully, then the operation shall fail without returning any instances.

595 The WBEM operations mapped to generic operations by a conformant WBEM protocol shall also either
596 succeed or fail, as described above.

597 **5.6 Preconditions and postconditions**

598 Each generic operation specifies a set of zero or more preconditions and a set of zero or more
599 postconditions.

600 Each precondition in the set needs to be satisfied for the operation to be able to succeed. If one or more
601 preconditions are not satisfied, the operation shall fail, indicating the unsatisfied precondition using a
602 generic error message from the set listed for the operation that describes the unsatisfied precondition.

603 A successful execution of the generic operation shall guarantee that all postconditions in the set are
604 satisfied.

605 **5.7 Generic error messages**

606 Each generic operation specifies a set of generic error messages. These generic error messages are
607 DMTF standard messages (see [DSP0228](#)) from the WBEM Operations Message Registry ([DSP8016](#)).
608 Each error message from this registry describes a particular error situation.

609 A conformant WBEM protocol shall support error handling in one or more of the following ways and shall
610 state in its WBEM protocol mapping which ways are supported:

- 611 • Return DMTF standard messages (also known as "extended error handling").
612 In this case, the WBEM operation shall return the generic error message defined for the generic
613 operation that matches the error situation, and may return additional error messages.
- 614 • Return CIM status codes.
615 In this case, the WBEM operation shall return the CIM status code stated in the generic error
616 message defined for the generic operation that matches the error situation. The CIM status
617 code values are stated in the definition of each generic message in [DSP8016](#).
- 618 • Return protocol-specific error representations.
619 In this case, the WBEM protocol mapping shall include a mapping of these protocol-specific
620 error representations to the generic messages defined in [DSP8016](#), and the WBEM operation
621 shall return the protocol-specific error representation corresponding to the generic error
622 message defined for the generic operation that matches the error situation.

623 The generic error messages specified for each generic operation have a requirement level defined in
624 context of that operation. The requirement level defines whether a conformant WBEM protocol has to
625 support the generic error message (in one or more of the ways defined above).

626 The allowable requirement levels for generic error messages in the context of a generic operation are:

627 **Mandatory**

628 Generic error messages designated as mandatory shall be supported by conformant WBEM
629 protocols if applicable to the WBEM protocol. They shall be supported as defined in the
630 description of the message.

631 **Conditional**

632 Generic error messages designated as conditional shall be supported by conformant WBEM
633 protocols if the specified condition is met and if applicable to the WBEM protocol. If supported,
634 they shall be supported as defined in the description of the message.

635 **Optional**

636 Generic error messages designated as optional may be supported by conformant WBEM
637 protocols if applicable to the WBEM protocol. If supported, they shall be supported as defined in
638 the description of the message.

639 Each generic operation designates one of its input parameters to be a "context parameter." The
640 messages defined in the WBEM Operations Message Registry ([DSP8016](#)) may include name and value
641 of the context parameter in order to provide information about the invocation context.

642 This specification does not define any order or precedence for generic error messages to be returned by
643 generic operations. This implies that the order in which the generic error messages are listed in the
644 description of each generic operation has no binding significance on the order in which a conformant
645 WBEM protocol would need to apply any tests to surface these errors, nor does the documented order
646 require a precedence of error messages. However, the order in which the generic error messages are
647 listed is meant to give some guidance about a typical order of precedence.

648 WBEM clients shall be prepared to deal with all generic error messages that are listed for a generic
649 operation.

650 **5.8 Consistency model**

651 This subclause defines consistency requirements for generic operations.

652 Conformant WBEM protocols shall conform to the rules defined in this subclause for the WBEM
653 operations to which the supported generic operations are mapped. WBEM protocols may define
654 additional constraints for WBEM operations.

655 This specification does not define responsibilities for detecting violations to these rules.

656 **5.8.1 Definition of ACID properties**

657 This subclause defines atomicity, consistency, isolation and durability (ACID) properties for use by
658 generic operations defined in this specification and by management profiles (see [DSP1001](#)).

659 Each generic operation defines requirements on its ACID properties. Management profiles that use
660 generic operations to state their operation requirements inherit these requirements on ACID properties
661 and may specify additional requirements. Profiles should not remove or weaken requirements on ACID
662 properties defined by generic operations.

663 **5.8.1.1 Atomicity**

664 Operations and methods are considered *atomic* if and only if their effects on the managed resources and
665 on CIM instances either occur completely or not at all.

666 Atomicity only applies to operations and methods that modify the managed resources or CIM instances
667 through the management interface.

668 **5.8.1.2 Update consistency**

669 Operations and methods are considered *update-consistent* if and only if the managed resources and CIM
670 instances are never left in an inconsistent state after a modification.

671 What constitutes a consistent state is defined in [DSP0004](#) and in management profiles.

672 Update consistency only applies to operations and methods that modify the managed resources or CIM
673 instances through the management interface.

674 5.8.1.3 Isolation

675 Operations and methods are considered *isolated* if and only if their results and their effects on the
676 managed resources and on CIM instances appear to be serialized with the results and effects of any
677 other operations and methods, as observed through the management interface.

678 Isolation applies to operations and methods that retrieve information through the management interface,
679 and to operations that modify the managed resources or CIM instances through the management
680 interface.

681 5.8.1.4 Durability

682 Operations and methods are considered *durable* if and only if their effects on the managed resources and
683 on CIM instances will not be undone, other than by some other action that may or may not be caused
684 through the profile defined management interface.

685 Durability only applies to operations and methods that modify the managed resources or CIM instances
686 through the management interface.

687 5.8.2 Time consistency within instance representations

688 The property values of an instance representation returned by any generic operation shall represent a
689 snapshot of the instance object that exists in the server.

690 If a WBEM protocol provides the capability to transfer an operation response in multiple parts, and a
691 particular instance representation is distributed over multiple parts of the response which are transferred
692 at different points in times, the property values of that instance representation still need to satisfy the time
693 consistency constraint.

694 5.8.3 Staleness of information returned

695 Conformant WBEM protocols should define that implementations should do a best effort to return the
696 most current information, as far as property values of instances and also the existence of instances are
697 concerned.

698 5.8.4 Isolation between operations

699 This specification defines no particular requirements regarding isolation between operations in addition to
700 the other consistency rules defined in 5.8.

701 For example, if an instance is deleted and after that another one is created, an enumeration operation
702 executed concurrently may consistently include the instance that got deleted just before that happened,
703 as well as the new instance after it got consistently created, hence returning a set of instances that never
704 existed at the same time. This example satisfies all consistency rules defined in this specification.
705 An example where other consistency rules determine the overall behavior is a GetInstance operation
706 executing concurrently with a

707 ModifyInstance operation on the same instance. The consistency rules defined in 5.8.2 require that this
708 GetInstance operation needs to return an instance representation that either has none or all of the
709 modifications requested by the ModifyInstance operation.

710 5.8.5 Duplicate return of CIM objects or object paths

711 Any generic operations returning CIM object representations or CIM object paths should not return
712 duplicate objects or duplicate object paths.

713 If duplicate objects or duplicate object paths are returned, WBEM clients should consider the last
714 occurrence of a duplicate object or duplicate object path in the sequence as the valid occurrence to work
715 with, and should ignore all other duplicate occurrences.

716 [DSP0004](#) requires that a CIM namespace in a WBEM server does not contain duplicate objects (i.e.,
717 instances, classes, qualifier types) at any point in time. However, given the rule above, the result set of a
718 generic operation may.

719 An example for a situation in which duplicate instances or instance paths might be returned is a sequence
720 of instance deletion and creation with the same key values concurrently to an enumeration operation, all
721 in the same namespace.

722 As a consequence, a WBEM server is not obliged to test for, correct or reject any duplicate objects or
723 object paths in the result set of an operation.

724 **5.8.6 Time consistency between returned CIM objects**

725 This specification does not mandate any time consistency between the CIM objects or CIM object paths
726 returned by generic operations.

727 For example, if a WBEM server processes an instance enumeration operation by contacting multiple
728 independent infrastructure components each of which contributes instances to the combined result set,
729 the result set may contain instance representations that represent different points in time.

730 However, the rule defined in 5.8.2 requires that consistency is maintained within each single instance
731 representation.

732 **5.8.7 Order of returned CIM objects**

733 For operations that do not support the specification of a sort order, the order of returned CIM objects is
734 implementation-dependent.

735 For example, if a WBEM server processes an instance enumeration operation by contacting multiple
736 independent infrastructure components each of which contributes instances to the combined result set,
737 the resulting order might be an arbitrary merge of the sequences of instances contributed by each
738 component.

739 WBEM protocols may define additional requirements on the order of returned CIM objects.

740 **5.8.8 Validity of returned object paths**

741 This specification does not mandate that object paths returned to a WBEM client are still valid by the time
742 the WBEM client attempts to use them in subsequent operations in order to address those objects.

743 For example: if a WBEM server returns an instance path and an operation then deletes the instance, a
744 subsequent attempt to get the instance using the returned instance path will fail.

745 **5.8.9 Effects of deleting an instance**

746 Deleting an instance may affect the overall consistency because other instances depend on the instance
747 to be deleted. Instances that depend on the instance to be deleted are called "dependent instances" in
748 this specification.

749 The behavior of operations that delete instances (such as *DeleteInstance*) cannot be defined in a
750 generally applicable way. The following options are available for defining the handling of the deletion of
751 an instance in the presence of dependent instances (e.g., in management profiles or in the CIM schema):

- 752 • **Delete propagation:** Delete any dependent instances implicitly along with the instance to be
753 deleted.

754 Specifications using this specification need to give particular consideration to circular
755 dependencies when defining rules for propagating deletion.

756 NOTE Such dependent instances may reside in a different namespace (which may reside in a different
757 WBEM server) than the instance to be deleted.

758 • **Rejection:** Reject the deletion of the instance to be deleted, leaving it to the WBEM client to
759 delete dependent instances first.

760 The following options are **not** available for defining the handling of the deletion of an instance in the
761 presence of dependent instances:

762 • **Deletion without propagation:** Delete the instance to be deleted but do not delete any
763 dependent instances. This would cause an inconsistent state in the model.

764 The following instances are considered dependent instances for this purpose:

765 • **Composition:** Instances associated to an instance to be deleted, via a composition where the
766 instance to be deleted is on the aggregate side.

767 The definition of the *Composition* qualifier in [DSP0004](#) requires that this case is handled by
768 propagating the deletion of the aggregate instance to any aggregated instances and their
769 composition instances.

770 • **Key propagation:** Instances of classes that have propagated keys (key properties exposing a
771 value of TRUE for the *Propagated* qualifier, i.e., weak instances) are considered dependents of
772 the instance from which the keys propagate (i.e., the strong instance).

773 The definition of the *Propagated* qualifier in [DSP0004](#) requires that this case is handled by
774 propagating the deletion of the strong instance to any weak instances and their association
775 instances.

776 • **Referencing associations:** Association instances that reference the instance to be deleted.

777 This case shall be handled with any or a combination of the following options:

778 – by propagating the deletion of the referenced instance to its referencing association
779 instance

780 – by rejecting the deletion of the referenced instance to be deleted.

781 • **Qualifier defined delete propagation:** Instances to be deleted as a result of *IfDelete* and
782 *Delete* qualifiers, as defined in [DSP0004](#).

783 Support of the *IfDelete* and *Delete* qualifiers by a WBEM server is optional, as defined in
784 [DSP0004](#).

785 This concept can be used to propagate deletion from an instance to its referencing association
786 instance, from an association instance to its referenced instances, and in combination also
787 between associated instances.

788 The definition of the *IfDelete* and *Delete* qualifiers in [DSP0004](#) requires that this case is handled
789 by propagating the deletion of an instance to which the *IfDelete* qualifier applies, to any
790 instances to which the corresponding *Delete* qualifier applies.

791 • **Multiplicity underflow:** Instances associated to an instance to be deleted via an association
792 with a minimum multiplicity (as defined with *Min* qualifier in the schema, or as constrained by
793 management profiles) larger than 0 on the reference to the instance to be deleted, if the deletion
794 would violate the minimum multiplicity that is required.

795 EXAMPLE: Association AB references class A with *Min* (2) and references class B. Therefore, each
796 instance of B is supposed to be associated via AB with least two instances of A. If an instance of A is to

797 be deleted, and there is only one other instance of A associated to the instance of B that is associated
 798 with the instance of A to be deleted, the minimum multiplicity would be violated by the deletion.

799 This case shall be handled with any or a combination of the following options:

- 800 – by propagating the deletion of the instance to be deleted to its associated instance defining
- 801 the multiplicity constraint, and the association instance.
- 802 – by rejecting the original deletion.

803 6 Generic operations

804 This clause defines the generic operations. They are listed in Table 1, grouped by their headings.

805 **Table 1 – List of generic operations**

Group	Generic Operation	Description
Instance operations	GetInstance	See 6.3.1
	DeleteInstance	See 6.3.2
	ModifyInstance	See 6.3.3
	CreateInstance	See 6.3.4
Direct instance enumeration operations	EnumerateInstances (deprecated)	See 6.4.1
	EnumerateInstanceNames (deprecated)	See 6.4.2
	Associators (deprecated)	See 6.4.3
	AssociatorNames (deprecated)	See 6.4.4
	References (deprecated)	See 6.4.5
	ReferenceNames (deprecated)	See 6.4.6
Pulled instance enumeration operations	OpenEnumerateInstances	See 6.5.3
	OpenEnumerateInstancePaths (deprecated)	See 6.5.4
	OpenAssociators	See 6.5.5
	OpenAssociatorPaths (deprecated)	See 6.5.6
	OpenReferences	See 6.5.7
	OpenReferencePaths (deprecated)	See 6.5.8
	OpenQueryInstances	See 6.5.9
	PullInstancesWithPath	See 6.5.11
	PullInstancePaths (deprecated)	See 6.5.12
	PullInstances	See 6.5.13
	CloseEnumeration	See 6.5.14
	EnumerationCount (deprecated)	See 6.5.15
Method invocation operations	InvokeMethod	See 6.6.1
	InvokeStaticMethod	See 6.6.2
Class operations	GetClass	See 6.7.1
	DeleteClass	See 6.7.2

Group	Generic Operation	Description
	ModifyClass	See 6.7.3
	CreateClass	See 6.7.4
Class enumeration operations	EnumerateClasses	See 6.8.1
	EnumerateClassNames	See 6.8.2
	AssociatorClasses	See 6.8.3
	AssociatorClassPaths	See 6.8.4
	ReferenceClasses	See 6.8.5
	ReferenceClassPaths	See 6.8.6
Qualifier type operations	GetQualifierType	See 6.9.1
	DeleteQualifierType	See 6.9.2
	ModifyQualifierType	See 6.9.3
	CreateQualifierType	See 6.9.4
	EnumerateQualifierTypes	See 6.9.5

806

807 **6.1 Description format**

808 The generic operations are described using the following format. Items in angle brackets (e.g., "<name>")
 809 need to be replaced by some other text, as described further down in this subclause.

810 **Purpose:**

811 <Short description of the purpose of the operation.>

812 **Operation input parameters:**

813

Generic Name	Generic Type	Requirement	Description
<diname>	<ditype>	<direq>	<Description of the operation parameter, including any conditions for requirement level Conditional> <The text "(Context Parameter)" for the parameter that is supposed to be displayed in messages, as defined in 5.7>
...

814

815 **Operation output parameters:**

816

Generic Name	Generic Type	Requirement	Description
<diname>	<ditype>	<direq>	<Description of the operation parameter, including any conditions for requirement level Conditional>
...

817

818 **Description:**

819 <A detailed description of the semantics of the operation including all conditions and behaviors
820 except those listed under Preconditions and Postconditions>

821 **Preconditions:**

822 • <List of additional preconditions for the operation, in plain text. Preconditions pertain to the state
823 before an operation gets invoked. They have nothing to do with the execution of the operation
824 or any effects the operation causes. They represent the conditions that are required to be met in
825 order for the operation to have a chance to execute successfully. Although not required for
826 preconditions, this specification uses "shall" to specify preconditions.>

827 **Postconditions:**

828 • <List of additional postconditions for the operation, in plain text. Postconditions describe the
829 state after an operation has been executed successfully. In other words, they represent the
830 guarantees an implementation needs to give in the case of successful execution.>

831 **Error messages:**

832

Message ID	Message Name	Requirement	Sources	Additional Description
<msgid>	<msgname>	<msgreq>	<msgsrc>	<Any description in addition to the description in the message registry>
...

833

834 The items in angle brackets that are not already described in the format above, have the following
835 meaning:

- 836 <diname> Generic name of the operation parameter.
- 837 <ditype> Generic type of the operation parameter, as defined in 5.4.
- 838 <direq> Requirement level of the operation parameter, as defined in 5.3.3.
- 839 <msgid> Message ID of the message, as defined in a DMTF message registry. The message
840 ID is the concatenation of the values of the XML attributes
841 MESSAGE/MESSAGE_ID@PREFIX and
842 MESSAGE/MESSAGE_ID@SEQUENCE_NUMBER.
- 843 <msgname> Message name of the message, as defined in a DMTF message registry. The
844 message name is the value of the XML attribute MESSAGE@NAME.
- 845 <msgreq> Requirement level of the message, as defined in 5.7.
- 846 <msgsrc> Sources of the message. One or more values may be specified. Valid values are:
847 Infrastructure – the message is implemented by the common infrastructure portion
848 of the WBEM server.
849 Class implem. – the message is implemented by the class specific portion of the
850 WBEM server.

851 The message sources information is a recommendation only, for implementations of
852 a WBEM server that distinguish between a common infrastructure portion (e.g.,
853 CIMOM) and class specific portion (e.g., providers).

854 **6.2 Common operation parameters for all operations**

855 This subclause defines commonly used operation parameters for the operations. The description of the
856 individual operations references these operation parameters as appropriate. However, not every
857 operation uses every one of these operation parameters.

858 **6.2.1 IncludeClassOrigin (partly deprecated)**

859 The *IncludeClassOrigin* operation input parameter controls whether class origin information is returned for
860 any element in any returned object. Class origin information indicates which class defines the element. In
861 case of multiple definitions due to overrides, class origin information indicates the leaf-most class in the
862 inheritance hierarchy that defines the element.

863 **Deprecated:** The *IncludeClassOrigin* operation parameter has been deprecated in version 1.1.0 of this
864 document for any instance operations; there is no replacement for it. The expectation is that clients do not
865 need to know the class origin information when retrieving instances.

866 Support for the *IncludeClassOrigin* operation parameter is conditional on support in the WBEM protocol
867 for client side control of returning class origin information.

868 If the WBEM protocol does not support client side control of returning class origin information, then the
869 *IncludeClassOrigin* operation parameter shall not be supported and class origin information shall be
870 included for any element in any object returned by the operation.

871 If the WBEM protocol supports client side control of returning class origin information, then the
872 *IncludeClassOrigin* operation parameter shall be supported. If the *IncludeClassOrigin* operation
873 parameter is TRUE, then class origin information shall be included for any element in any object returned
874 by the operation. If the *IncludeClassOrigin* operation parameter is FALSE, then class origin information
875 shall not be included for any element in any object returned by the operation.

876 For operations returning instance representations, the elements are properties only (more precisely, their
877 values). For operations returning class representations, the elements are properties and methods (more
878 precisely, their definitions).

879 **6.2.2 IncludeQualifiers**

880 The *IncludeQualifiers* operation input parameter controls whether qualifier values are returned for any
881 returned CIM element in any returned class of a class operation.

882 Support for the *IncludeQualifiers* operation parameter in a conformant WBEM protocol is mandatory.

883 If *IncludeQualifiers* is TRUE, then any returned class and any returned CIM element within each returned
884 class shall contain qualifier values for those qualifiers that have a value different from the default value
885 defined in the declaration of the qualifier type. Any other qualifier values should not be included.

886 **NOTE** In order to inspect the scope and default value of any qualifiers that are not included in the returned class,
887 a WBEM client can use operation *EnumerateQualifierTypes* to retrieve the qualifier type declarations that exist in a
888 namespace.

889 If *IncludeQualifiers* is FALSE, then any returned class and any returned CIM element within each returned
890 class shall not contain any qualifier values.

891 **6.2.3 <element>List**

892 The operation output parameters *InstanceList*, *InstancePathList*, *ClassList*, *ClassPathList*, and
 893 *QualifierTypeList* contain a sequence of elements, and are referred to as the *result set* of the operation.

894 The sequence is ordered in the sense that there is a relation of "before" and "after" between elements in
 895 the sequence and the sequence has a beginning and an end. However, this does not imply that the
 896 sequence is sorted according to some criteria.

897 Clause 5.8 defines rules for dealing with duplicate objects or duplicate object paths in the result set of an
 898 operation.

899 **6.3 Instance operations**

900 This subclause defines server operations that target a single instance, or create an instance.

901 **6.3.1 GetInstance**

902 **Purpose:**

903 Retrieves an instance.

904 **Operation input parameters:**

905

Generic Name	Generic Type	Requirement	Description
InstancePath	InstancePath	Mandatory	Instance path of the instance to be retrieved (Context Parameter)
IncludeClassOrigin	boolean	Conditional	Deprecated: Indicates whether class origin information for any returned properties is to be included, as defined in 6.2.1 Condition: WBEM protocol supports client side control of returning class origin information.
IncludedProperties	PropertyName []	Optional	NULL, or unordered set of property names, acting as a restricting filter on the properties included in the returned instance

906

907 **Operation output parameters:**

908

Generic Name	Generic Type	Requirement	Description
Instance	InstanceSpecification	Mandatory	Representation of the retrieved instance

909

910 **Description:**

911 The *GetInstance* operation retrieves a representation of the instance referenced by *InstancePath*.

912 As defined in the description of the *InstancePath* type, the instance path of the instance to be
 913 retrieved is interpreted in a non-polymorphic way, i.e., it references the specified instance only and
 914 does not include any instances with the same key values in subclasses.

915 The set of properties to be included in the retrieved instance shall be determined using the following
916 algorithm:

- 917 • Initially, the set of properties to be included is the set of properties exposed by the creation
918 class of the instance. This includes all the duplicates of any duplicate non-overridden
919 properties.
- 920 • If the *IncludedProperties* operation input parameter is supported by the WBEM protocol
921 and if its value is not NULL, it acts as a restricting filter on the properties to be included in
922 the returned instance representation such that any properties exposed by the creation
923 class of the instance that are not named in that operation parameter are removed from the
924 set of properties to be included. Any duplicate or invalid property names in the
925 *IncludedProperties* operation input parameter shall be ignored. A non-NULL empty
926 *IncludedProperties* list removes all properties from the set of properties to be included.
- 927 • Conformant WBEM protocols may specify rules that cause properties with a value of NULL
928 to be removed from the set of properties to be included.

929 Preconditions:

- 930 • The instance referenced by *InstancePath* shall exist. If it does not exist, the operation shall fail,
931 indicating WIPG0213.
- 932 • The creation class of the instance referenced by *InstancePath* shall exist. If it does not exist, the
933 operation shall fail, indicating WIPG0214.
- 934 • The namespace of the instance referenced by *InstancePath* shall exist. If it does not exist, the
935 operation shall fail, indicating WIPG0204.

936 Postconditions:

- 937 • The instance representation shall have been returned with the properties as defined in the
938 Description paragraph for this operation.
- 939 • Requirements on ACID properties:
 - 940 – Atomicity: N/A
 - 941 – Update Consistency: N/A
 - 942 – Isolation: Required
 - 943 – Durability: N/A

944 Error messages:

945

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM server is shutting down	Optional	Infrastructure	
WIPG0240	WBEM server limits are exceeded	Optional	Infrastructure, class implem.	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM server infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure, class implem.	
WIPG0214	Class not found	Mandatory	Infrastructure	
WIPG0228	Operation not supported by class implementation	Mandatory	Class implem.	
WIPG0213	Instance not found	Mandatory	Class implem.	
WIPG0243	Timeout	Optional	Infrastructure, class implem.	
WIPG0227	Other failure	Optional	Infrastructure, class implem.	

946

947 **6.3.2 DeleteInstance**

948 **Purpose:**

949 Deletes an instance.

950 **Operation input parameters:**

951

Generic Name	Generic Type	Requirement	Description
InstancePath	InstancePath	Mandatory	Instance path of the instance to be deleted (Context Parameter)

952

953 **Operation output parameters:**

954 None.

955 **Description:**

956 The *DeleteInstance* operation deletes the instance referenced by *InstancePath*.

957 The existence of other instances may depend on the instance to be deleted. There are multiple types
 958 of dependent instances, and multiple options to handle such dependent instances, as defined in
 959 5.8.9.

960 **NOTE** Any dependent instances that are deleted may reside in a different namespace (which may reside in a
 961 different WBEM server) than the instance referenced by *InstancePath*.

962 In case of error, the consistency requirements defined in [DSP0004](#) cannot be guaranteed, but should
 963 be attempted to be satisfied in a best effort approach. Such an approach may be to delete non-
 964 dependent instances first. In case of error, only a subset of the instances to be deleted may have
 965 been deleted, but each instance shall have either been deleted completely or not at all.

966 The effects of the deletion of any instances on managed resources shall be defined elsewhere. For
 967 example, a management profile may define that the lifecycle of the instance is coupled with the

968 lifecycle of some underlying managed resource, and that this resource shall be deleted when the
969 instance is deleted.

970 **Preconditions:**

- 971 • The instance referenced by *InstancePath* shall exist. If it does not exist, the operation shall fail,
972 indicating WIPG0213.
- 973 • The creation class of the instance referenced by *InstancePath* shall exist. If it does not exist, the
974 operation shall fail, indicating WIPG0214.
- 975 • The namespace of the instance referenced by *InstancePath* shall exist. If it does not exist, the
976 operation shall fail, indicating WIPG0204.

977 **Postconditions:**

- 978 • The instance referenced by *InstancePath* shall have been deleted.
- 979 • Any implicit deletions of dependent instances shall have happened, as defined in 5.8.9.
- 980 • Any effects of the deletion of all of these instances on any managed resources shall have
981 happened.
- 982 • The consistency requirements defined in [DSP0004](#) shall be satisfied for any instances related to
983 the deleted instances.
- 984 • Requirements on ACID properties:
 - 985 – Atomicity: Required, if dependent instances are handled by rejection, as defined in 5.8.9.
986 Recommended, if dependent instances are handled by delete propagation, as defined in
987 5.8.9.
 - 988 – Update Consistency: Required, if dependent instances are handled by rejection, as defined
989 in 5.8.9. Recommended, if dependent instances are handled by delete propagation, as
990 defined in 5.8.9.
 - 991 – Isolation: Required, if dependent instances are handled by rejection, as defined in 5.8.9.
992 Recommended, if dependent instances are handled by delete propagation, as defined in
993 5.8.9.
 - 994 – Durability: Required.

995 **Error messages:**

996

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM server is shutting down	Optional	Infrastructure	
WIPG0240	WBEM server limits are exceeded	Optional	Infrastructure, class implem.	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM server infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure, class implem.	
WIPG0214	Class not found	Mandatory	Infrastructure	
WIPG0228	Operation not supported by class implementation	Mandatory	Class implem.	
WIPG0213	Instance not found	Mandatory	Class implem.	
WIPG0246	Instance cannot be deleted due to referencing association	Optional	Class implem.	
WIPG0247	Instance cannot be deleted due to multiplicity underflow	Optional	Class implem.	
WIPG0243	Timeout	Optional	Infrastructure, class implem.	
WIPG0227	Other failure	Optional	Infrastructure, class implem.	

997

998 **6.3.3 ModifyInstance**

999 **Purpose:**

1000 Changes property values of a given instance.

1001 **Operation input parameters:**

1002

Generic Name	Generic Type	Requirement	Description
InstancePath	InstancePath	Mandatory	Instance path of the instance to be modified (Context Parameter)
ModifiedInstance	InstanceSpecification	Mandatory	Representation of the modified instance, specifying the new property values
IncludedProperties	PropertyName []	Optional	NULL, or unordered set of property names, acting as a restricting filter on the properties to be modified

1003

1004 **Operation output parameters:**

1005 None.

1006 **Description:**

1007 The *ModifyInstance* operation changes property values of the instance referenced by *InstancePath*.

- 1008 The set of properties to be changed shall be determined using the following algorithm:
- 1009 • Initially, the set of properties to be changed is the set of properties specified in
1010 *ModifiedInstance*.
 - 1011 • If the *IncludedProperties* operation input parameter is supported by the WBEM protocol
1012 and if its value is not NULL, it acts as a restricting filter on the properties to be changed
1013 such that any properties exposed by the creation class of the instance that are not named
1014 in that operation parameter are removed from the set of properties to be changed. Any
1015 duplicate or invalid property names in the *IncludedProperties* operation input parameter
1016 shall be ignored. A non-NULL empty *IncludedProperties* list removes all properties from
1017 that set.
 - 1018 • Any key properties and non-modifiable properties are removed from the set of properties to
1019 be changed. As a result, specifying such properties in *ModifiedInstance* or
1020 *IncludedProperties* does not cause an error.
- 1021 NOTE The modifiability of properties can be defined in the schema and in management profiles.
- 1022 Conformant WBEM protocols may restrict *ModifiedInstance* to specify all properties exposed by the
1023 creation class of the instance referenced by *InstancePath*.

1024 Preconditions:

- 1025 • The instance referenced by *InstancePath* shall exist. If it does not exist, the operation shall fail,
1026 indicating WIPG0213.
- 1027 • The creation class of the instance referenced by *InstancePath* shall exist. If it does not exist, the
1028 operation shall fail, indicating WIPG0214.
- 1029 • The namespace of the instance referenced by *InstancePath* shall exist. If it does not exist, the
1030 operation shall fail, indicating WIPG0204.
- 1031 • The creation class of *ModifiedInstance* shall be the creation class of the instance referenced by
1032 *InstancePath* or a superclass of that class. If this is not satisfied, the operation shall fail,
1033 indicating WIPG0208.
- 1034 • Any properties specified in *ModifiedInstance* shall be from the set of properties exposed by the
1035 creation class of *ModifiedInstance*. If this is not satisfied, the operation shall fail, indicating
1036 WIPG0208.

1037 Postconditions:

- 1038 • The values of the properties shall have been modified as defined in the Description paragraph
1039 for this operation.
- 1040 • The values of key properties and non-modifiable properties shall not have been modified.
- 1041 • Other properties may have changed as a result of side effects of changing properties, behavior
1042 defined in referencing specifications, or volatility of properties.
- 1043 • The consistency requirements defined in [DSP0004](#) shall be satisfied for the modified instance.
- 1044 • Requirements on ACID properties:
 - 1045 – Atomicity: Required
 - 1046 – Update Consistency: Required
 - 1047 – Isolation: Required
 - 1048 – Durability: Required

1049 **Error messages:**

1050

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM server is shutting down	Optional	Infrastructure	
WIPG0240	WBEM server limits are exceeded	Optional	Infrastructure, class implem.	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM server infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure, class implem.	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure, class implem.	
WIPG0214	Class not found	Mandatory	Infrastructure	
WIPG0228	Operation not supported by class implementation	Mandatory	Class implem.	
WIPG0213	Instance not found	Mandatory	Class implem.	
WIPG0220	No such property	Mandatory	Class implem.	
WIPG0243	Timeout	Optional	Infrastructure, class implem.	
WIPG0227	Other failure	Optional	Infrastructure, class implem.	

1051

1052 **6.3.4 CreateInstance**

1053 **Purpose:**

1054 Creates an instance of a given class.

1055 **Operation input parameters:**

1056

Generic Name	Generic Type	Requirement	Description
ClassPath	ClassPath	Mandatory	Class path of the creation class of the instance to be created (Context Parameter)
NewInstance	InstanceSpecification	Optional	Instance representation specifying the initial property values for the instance to be created

1057

1058 **Operation output parameters:**

1059

Generic Name	Generic Type	Requirement	Description
InstancePath	InstancePath	Mandatory	Instance path of the new instance

1060

1061 **Description:**

1062 The *CreateInstance* operation creates an instance of the creation class referenced by *ClassPath* in
 1063 the same namespace as that creation class and returns the instance path of the new instance.

1064 The creation class is interpreted in a non-polymorphic way; that is, the creation class of the newly
 1065 created instance shall be specified creation class (and not a subclass thereof).

1066 The newly created instance shall have all properties exposed by the creation class referenced by
 1067 *ClassPath*.

1068 For each property, its initial value in the new instance shall be determined as follows:

- 1069 • If the *NewInstance* operation input parameter is supported, and if the property is included
 1070 in *NewInstance*, its value is used as the initial value. That is also the case if that value is
 1071 NULL.
- 1072 • Else, if an initialization constraint is defined for the property (that is, through the class-
 1073 defined property default value, a use of the PropertyConstraint qualifier, or by a
 1074 management profile), a value satisfying that constraint is used as the initial value.
- 1075 • Else, the initial value is implementation-defined.

1076 Key properties and non-writeable properties included in *NewInstance* shall be treated like any other
 1077 properties; the creation of an instance does not have the restrictions a subsequent modification has.

1078 Volatile properties may change their values immediately after the instance has been created.

1079 Instance creation based upon input data other than initial property values can be done using CIM
 1080 methods. For example, creation of an instance of *CIM_ComputerSystem* representing a virtual
 1081 computer system could be done using a *CreateVirtualComputerSystem()* method taking a higher-
 1082 level specification of the virtual computer system as input.

1083 Other instances may come into existence implicitly during the course of processing the
 1084 *CreateInstance* operation. As defined in [DSP1001](#), management profiles may specify the rules for
 1085 such implicitly created instances.

1086 Any such implicitly created instances may reside in the same or a different namespace (which may
 1087 reside in a different WBEM server) than the namespace of the creation class referenced by
 1088 *ClassPath*.

1089 In case of error, the consistency requirements defined in [DSP0004](#) should be attempted to be
 1090 satisfied in a best effort approach. In case of error, only a subset of the instances to be created may
 1091 have been created, but each instance shall have either been created completely or not at all.

1092 As defined in [DSP1001](#), management profiles may specify the effects of the creation of instances on
 1093 managed resources. For example, a management profile may define that the lifecycle of the instance
 1094 is coupled with the lifecycle of some underlying managed resource, and that this resource shall be
 1095 created when the instance is created.

1096 **Preconditions:**

- 1097 • The instance to be created shall not exist in the namespace specified by *ClassPath*. If this is not
1098 satisfied, the operation shall fail, indicating WIPG0216.
- 1099 • The class referenced by *ClassPath* shall exist. If it does not exist, the operation shall fail,
1100 indicating WIPG0214.
- 1101 • The namespace of the class referenced by *ClassPath* shall exist. If it does not exist, the
1102 operation shall fail, indicating WIPG0204.
- 1103 • The creation class of *NewInstance* shall be the class referenced by *ClassPath* or a superclass
1104 of that class. If this is not satisfied, the operation shall fail, indicating WIPG0208.
- 1105 • Any properties specified in *NewInstance* shall be from the set of properties exposed by the
1106 class referenced by *ClassPath*. If this is not satisfied, the operation shall fail, indicating
1107 WIPG0208.
- 1108 • If the schema definition of the class referenced by *ClassPath* or any implemented management
1109 profiles require that *NewInstance* includes a property, but that property is not included in
1110 *NewInstance*, the operation shall fail, indicating WIPG0249.
- 1111 • If the schema definition of the class referenced by *ClassPath* or any implemented management
1112 profiles require that *NewInstance* does not include a property, but that property is included in
1113 *NewInstance*, the operation shall fail, indicating WIPG0249.

1114 **Postconditions:**

- 1115 • The instance shall have been created as defined in the Description paragraph for this operation.
- 1116 • Any management profile defined implicit creations of other instances shall have happened.
- 1117 • Any management profile defined effects of the creation of all of these instances on any
1118 managed resources shall have happened.
- 1119 • Requirements on ACID properties:
 - 1120 – Atomicity: Required
 - 1121 – Update Consistency: Required
 - 1122 – Isolation: Required
 - 1123 – Durability: Required

1124 **Error messages:**

1125

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM server is shutting down	Optional	Infrastructure	
WIPG0240	WBEM server limits are exceeded	Optional	Infrastructure, class implem.	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM server infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure, class implem.	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure, class implem.	
WIPG0214	Class not found	Mandatory	Infrastructure	
WIPG0228	Operation not supported by class implementation	Mandatory	Class implem.	
WIPG0216	Instance already exists	Mandatory	Class implem.	
WIPG0243	Timeout	Optional	Infrastructure, class implem.	
WIPG0227	Other failure	Optional	Infrastructure, class implem.	

1126

1127 6.4 Direct instance enumeration operations

1128 This subclause defines server operations that enumerate instances and return their instance
1129 representations or instance paths directly as a result of the operation.

1130 6.4.1 EnumerateInstances (deprecated)

1131 Purpose:

1132 Enumerate the instances of a given class and return their instance representations and instance
1133 paths.

1134 Operation input parameters:

1135

Generic Name	Generic Type	Requirement	Description
EnumClassPath	ClassPath	Mandatory	Class path of the class whose instances are to be enumerated (Context Parameter)
IncludeClassOrigin	boolean	Conditional	Deprecated: Indicates whether class origin information for any returned properties is to be included, as defined in 6.2.1 Condition: WBEM protocol supports client side control of returning class origin information.
IncludedProperties	PropertyName []	Optional	NULL, or unordered set of property names to be included, acting as a restricting filter on the properties included in the returned instance representations
ExcludeSubclassProperties	boolean	Optional	Indicates whether properties added by subclasses of the specified class are to be excluded, acting as a restricting filter on the properties included in the returned instance representations

1136

1137 **Operation output parameters:**

1138

Generic Name	Generic Type	Requirement	Description
InstanceList	InstanceSpecificationWithPath []	Mandatory	Sequence of the returned instance representations and instance paths

1139

1140 **Description:**

1141 The *EnumerateInstances* operation enumerates all instances of the class referenced by
1142 *EnumClassPath*, including instances of its subclasses, and returns their instance representations
1143 and instance paths.

1144 The *EnumerateInstances* operation has been deprecated in version 1.1.0 of this document. Use
1145 *OpenEnumerateInstances* instead (see 6.5.3).

1146 All of the enumerated instances shall exist in the same namespace as the class referenced by
1147 *EnumClassPath*.

1148 An instance is included in the result set if and only if it exists in the namespace of the class
1149 referenced by *EnumClassPath*, and its creation class is the class referenced by *EnumClassPath* or a
1150 subclass of that class.

1151 The result set should not contain any duplicate instances, as defined in 5.8.4. Because the result set
1152 contains only instances that exist in the same namespace, a determination of duplicate instances (for
1153 example by the client) can be done on the basis of their model paths only.

1154 The set of properties to be included in any instance representations in the result set shall be
1155 determined using the following algorithm:

- 1156 • Initially, the set of properties to be included is the set of properties exposed by the creation
1157 class of the instance. This includes all the duplicates of any duplicate non-overridden
1158 properties.
- 1159 • If the *IncludedProperties* operation input parameter is supported by the WBEM protocol
1160 and if its value is not NULL, it acts as a restricting filter on the properties to be included in
1161 the returned instance representations such that any properties exposed by the creation
1162 class of the instance that are not named in that operation parameter are removed from the
1163 set of properties to be included. Any duplicate or invalid property names in the
1164 *IncludedProperties* operation input parameter shall be ignored. A non-NULL empty
1165 *IncludedProperties* list removes all properties from the set of properties to be included.
- 1166 • If the *ExcludeSubclassProperties* operation input parameter is supported by the WBEM
1167 protocol and if its value is TRUE, it acts as a restricting filter on the properties to be
1168 included in the returned instance representations such that any properties not exposed by
1169 the class referenced by *EnumClassPath* are removed from the set of properties to be
1170 included. In other words, the set of properties is restricted to the properties exposed by the
1171 enumeration class.
- 1172 • Conformant WBEM protocols may specify rules that cause properties with a value of NULL
1173 to be removed from the set of properties to be included.

1174 **Preconditions:**

- 1175 • The class referenced by *EnumClassPath* shall exist. If it does not exist, the operation shall fail,
1176 indicating WIPG0214.
- 1177 • The namespace of the class referenced by *EnumClassPath* shall exist. If it does not exist, the
1178 operation shall fail, indicating WIPG0204.

1179 **Postconditions:**

- 1180 • The instance representations and instance paths shall have been returned as described in the
1181 Description paragraph for this operation.
- 1182 • Requirements on ACID properties:
- 1183 – Atomicity: N/A
- 1184 – Update Consistency: N/A
- 1185 – Isolation: Required at the level of single instances, as defined in 5.8.
- 1186 – Durability: N/A

1187 **Error Messages:**

1188

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM server is shutting down	Optional	Infrastructure	
WIPG0240	WBEM server limits are exceeded	Optional	Infrastructure, class implem.	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM server infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure, class implem.	
WIPG0214	Class not found	Mandatory	Infrastructure	
WIPG0228	Operation not supported by class implementation	Mandatory	Class implem.	
WIPG0243	Timeout	Optional	Infrastructure, class implem.	
WIPG0227	Other failure	Optional	Infrastructure, class implem.	

1189

1190 **6.4.2 EnumerateInstanceNames (deprecated)**

1191 **Purpose:**

1192 Enumerate the instances of a given class and return their instance paths.

1193 **Operation input parameters:**

1194

Generic Name	Generic Type	Requirement	Description
EnumClassPath	ClassPath	Mandatory	Class path of the class whose instances are to be enumerated (Context Parameter)

1195

1196 **Operation output parameters:**

1197

Generic Name	Generic Type	Requirement	Description
InstancePathList	InstancePath []	Mandatory	Sequence of returned instance paths

1198

1199 **Description:**

1200 The *EnumerateInstanceNames* operation enumerates the instances of the class referenced by
1201 *EnumClassPath* and returns their instance paths.

1202 The *EnumerateInstanceNames* operation has been deprecated in version 1.1.0 of this document.
1203 Use *OpenEnumerateInstancePaths* instead (see 6.5.4).

1204 An instance is included in the result set if and only if it exists in the namespace of the class
1205 referenced by *EnumClassPath* and its creation class is the class referenced by *EnumClassPath* or a
1206 subclass of that class.

1207 The result set should not contain any duplicate instances, as defined in 5.8.4. Because the result set
1208 contains only instances that exist in the same namespace, a determination of duplicate instances (for
1209 example by the client) can be done on the basis of their model paths only.

1210 **Preconditions:**

- 1211 • The class referenced by *EnumClassPath* shall exist. If it does not exist, the operation shall fail,
1212 indicating WIPG0214.
- 1213 • The namespace of the class referenced by *EnumClassPath* shall exist. If it does not exist, the
1214 operation shall fail, indicating WIPG0204.

1215 **Postconditions:**

- 1216 • The instance paths shall have been returned as described in the Description paragraph for this
1217 operation.
- 1218 • Requirements on ACID properties:
 - 1219 – Atomicity: N/A
 - 1220 – Update Consistency: N/A

- 1221 – Isolation: Required at the level of single instances, as defined in 5.8.
- 1222 – Durability: N/A

1223 **Error messages:**

1224

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM server is shutting down	Optional	Infrastructure	
WIPG0240	WBEM server limits are exceeded	Optional	Infrastructure, class implem.	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM server infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure, class implem.	
WIPG0214	Class not found	Mandatory	Infrastructure	
WIPG0228	Operation not supported by class implementation	Mandatory	Class implem.	
WIPG0243	Timeout	Optional	Infrastructure, class implem.	
WIPG0227	Other failure	Optional	Infrastructure, class implem.	

1225

1226 **6.4.3 Associators (deprecated)**

1227 **Purpose:**

- 1228 Enumerate the instances that are associated with a given source instance and return their instance
- 1229 representations and instance paths.

1230 **Operation input parameters:**

1231

Generic Name	Generic Type	Requirement	Description
SourceInstancePath	InstancePath	Mandatory	Instance path of the source instance (Context Parameter)
AssociationClassName	ClassName	Mandatory	NULL, or name of the association class, acting as a restricting filter on the returned instances

Generic Name	Generic Type	Requirement	Description
AssociatedClassName	ClassName	Mandatory	NULL, or name of the associated class on any far end of the association, acting as a restricting filter on the returned instances
SourceRoleName	PropertyName	Mandatory	NULL, or name of the role on the source end of the association, acting as a restricting filter on the returned instances
AssociatedRoleName	PropertyName	Mandatory	NULL, or name of the role on any far end of the association, acting as a restricting filter on the returned instances
IncludeClassOrigin	boolean	Conditional	Deprecated: Indicates whether class origin information for any returned properties is to be included, as defined in 6.2.1 Condition: WBEM protocol supports client side control of returning class origin information.
IncludedProperties	PropertyName []	Optional	NULL, or unordered set of property names to be included, acting as a restricting filter on the properties included in the returned instances
ExcludeSubclassProperties	boolean	Optional	Indicates whether properties added by subclasses of the associated class are to be excluded, acting as a restricting filter on the properties included in the returned instances

1232

1233 **Operation output parameters:**

1234

Generic Name	Generic Type	Requirement	Description
InstanceList	InstanceSpecificationWithPath []	Mandatory	Sequence of the returned instance representations and instance paths

1235

1236 **Description:**

1237 The *Associators* operation enumerates the instances that are associated with the source instance
1238 referenced by *SourceInstancePath* and returns their instance representations and instance paths.

1239 The *Associators* operation has been deprecated in version 1.1.0 of this document. Use
1240 *OpenAssociators* instead (see 6.5.5).

1241 The set of associated instances to be enumerated shall be determined using the following algorithm:

- 1242 • Initially, the set of instances to be enumerated is the set of all instances associated to the
1243 source instance referenced by *SourceInstancePath*. The associations may be instances of
1244 different association classes. If the source instance does not exist, the operation shall
1245 succeed with an empty result set (even when its creation class does not exist). However, if
1246 the namespace of the source instance does not exist, the operation shall fail, indicating
1247 WIPG0204.

1248 The result set should not contain any duplicate instances, as defined in 5.8.4. However,
1249 different far ends may reference the same instance, and in such cases, the instance shall
1250 be contained in the result set once for each such reference.

- 1251
- 1252
- 1253
- 1254
- 1255
- 1256
- 1257
- If the *AssociationClassName* operation input parameter is not NULL, it acts as a restricting filter on the instances to be enumerated such that each instance that is associated with the source instance using an association whose creation class or one of its superclasses does not have the name specified in *AssociationClassName*, is removed from the set of instances to be enumerated. There shall be no validity checking performed for the *AssociationClassName* operation input parameter; if the specified class does not exist, the operation shall succeed with an empty result (because the filter did not match).
- 1258
- 1259
- 1260
- 1261
- 1262
- 1263
- 1264
- If the *AssociatedClassName* operation input parameter is not NULL, it acts as a restricting filter on the instances to be enumerated such that each instance whose creation class or one of its superclasses does not have the name specified in *AssociatedClassName*, is removed from the set of instances to be enumerated. There shall be no validity checking performed for the *AssociatedClassName* operation input parameter; if the specified class does not exist, the operation shall succeed with an empty result (because the filter did not match).
- 1265
- 1266
- NOTE Specifying a non-NULL value for *AssociatedClassName* ensures that the returned instances have the class specified in *AssociatedClassName* as a common superclass.
- 1267
- 1268
- 1269
- 1270
- 1271
- 1272
- 1273
- If the *SourceRoleName* operation input parameter is not NULL, it acts as a restricting filter on the instances to be enumerated such that each instance that is associated with the source instance using an association class that has a role name on the source end that is not the role name specified in *SourceRoleName*, is removed from the set of instances to be enumerated. There shall be no validity checking performed for the *SourceRoleName* operation input parameter; if the specified role does not exist, the operation shall succeed with an empty result (because the filter did not match).
- 1274
- 1275
- 1276
- 1277
- 1278
- 1279
- 1280
- If the *AssociatedRoleName* operation input parameter is not NULL, it acts as a restricting filter on the instances to be enumerated such that each instance that is associated with the source instance using an association class that has a role name on the end referencing that instance that is not the role name specified in *AssociatedRoleName*, is removed from the set of instances to be enumerated. There shall be no validity checking performed for the *AssociatedRoleName* operation input parameter; if the specified role does not exist, the operation shall succeed with an empty result (because the filter did not match).
- 1281
- 1282
- The set of properties to be included in each returned instance representation shall be determined using the following algorithm:
- 1283
- 1284
- 1285
- Initially, the set of properties to be included is the set of properties exposed by the creation class of the instance. This includes all the duplicates of any duplicate non-overridden properties.
- 1286
- 1287
- 1288
- 1289
- 1290
- 1291
- 1292
- If the *IncludedProperties* operation input parameter is supported by the WBEM protocol and if its value is not NULL, it acts as a restricting filter on the properties to be included in the returned instance representations such that any properties exposed by the creation class of the instance that are not named in that operation parameter are removed from the set of properties to be included. Any duplicate or invalid property names in the *IncludedProperties* operation input parameter shall be ignored. A non-NULL empty *IncludedProperties* list removes all properties from the set of properties to be included.
- 1293
- 1294
- 1295
- 1296
- 1297
- If the *ExcludeSubclassProperties* operation input parameter is supported by the WBEM protocol and if its value is TRUE, it acts as a restricting filter on the properties to be included in the returned instance representations such that any properties not exposed by the class specified in *AssociatedClassName* are removed from the set of properties to be included.
- 1298
- 1299
- Conformant WBEM protocols may specify rules that cause properties with a value of NULL to be removed from the set of properties to be included.

1300 **Preconditions:**

- 1301 • The namespace of the source instance referenced by *SourceInstancePath* shall exist. If it does
1302 not exist, the operation shall fail, indicating WIPG0204.
- 1303 • The *IncludedProperties* operation parameter, if supported by the WBEM protocol, shall only be
1304 specified with a non-NULL value if the *AssociatedClassName* operation input parameter is also
1305 non-NULL. If this is not satisfied, the operation shall fail, indicating WIPG0208.
- 1306 • The *ExcludeSubclassProperties* operation parameter, if supported by the WBEM protocol, shall
1307 only be specified with a TRUE value if the *AssociatedClassName* operation input parameter is
1308 non-NULL. If this is not satisfied, the operation shall fail, indicating WIPG0208.
- 1309 • The namespace of any returned instance paths shall exist. If it does not exist, the operation may
1310 fail, indicating WIPG0204. Note that cross-namespace association traversals may return
1311 instance paths in a server or namespace that is different from the server or namespace of the
1312 source instance.
- 1313 • The creation class of any returned instance paths shall exist in their namespace. If it does not
1314 exist, the operation may fail, indicating WIPG0214.

1315 **Postconditions:**

- 1316 • The instance representations and instance paths shall have been returned as described in the
1317 Description paragraph for this operation.
- 1318 • Requirements on ACID properties:
1319 – Atomicity: N/A
1320 – Update Consistency: N/A
1321 – Isolation: Required at the level of single instances, as defined in 5.8.
1322 – Durability: N/A

1323 **Error Messages:**

1324

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM server is shutting down	Optional	Infrastructure	
WIPG0240	WBEM server limits are exceeded	Optional	Infrastructure, class implem.	
WIPG0204	Namespace not found	Mandatory	Infrastructure	For input namespace
WIPG0203	Operation not supported by WBEM server infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure, class implem.	

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0228	Operation not supported by class implementation	Mandatory	Class implem.	
WIPG0204	Namespace not found	Optional	Infrastructure	For namespace of returned instance paths
WIPG0214	Class not found	Optional	Infrastructure	
WIPG0243	Timeout	Optional	Infrastructure, class implem.	
WIPG0227	Other failure	Optional	Infrastructure, class implem.	

1325

1326 NOTE Version 1.1 of this specification removed WIPG0213 (Instance not found) from this table.

1327 **6.4.4 AssociatorNames (deprecated)**1328 **Purpose:**1329 Enumerate instances that are associated with a given source instance and return their instance
1330 paths.1331 **Operation input parameters:**

1332

Generic Name	Generic Type	Requirement	Description
SourceInstancePath	InstancePath	Mandatory	Instance path of the source instance (Context Parameter)
AssociationClassName	ClassName	Mandatory	NULL, or name of the association class, acting as a restricting filter on the returned instances
AssociatedClassName	ClassName	Mandatory	NULL, or name of the associated class on any far end of the association, acting as a restricting filter on the returned instances
SourceRoleName	PropertyName	Mandatory	NULL, or name of the role on the source end of the association, acting as a restricting filter on the returned instances
AssociatedRoleName	PropertyName	Mandatory	NULL, or name of the role on any far end of the association, acting as a restricting filter on the returned instances

1333

1334 **Operation output parameters:**

1335

Generic Name	Generic Type	Requirement	Description
InstancePathList	InstancePath []	Mandatory	Sequence of the returned instance paths

1336

1337 **Description:**1338 The *AssociatorNames* operation enumerates instances that are associated with the source instance
1339 referenced by *SourceInstancePath* and returns their instance paths.

- 1340 The *AssociatorNames* operation has been deprecated in version 1.1.0 of this document. Use
1341 *OpenAssociators* instead (see 6.5.5).
- 1342 The set of instances to be enumerated shall be determined using the following algorithm:
- 1343 • Initially, the set of instances to be enumerated is the set of all instances associated to the
1344 source instance referenced by *SourceInstancePath*. The associations may be instances of
1345 different association classes. If the source instance does not exist, the operation shall
1346 succeed with an empty result set (even when its creation class does not exist). However, if
1347 the namespace of the source instance does not exist, the operation shall fail, indicating
1348 WIPG0204.
 - 1349 The result set should not contain any duplicate instances, as defined in 5.8.4. However,
1350 different association instances may reference the same instance on one of their far ends,
1351 and in such cases, the instance shall be contained in the result set once for each such
1352 reference.
 - 1353 • If the *AssociationClassName* operation input parameter is not NULL, it acts as a restricting
1354 filter on the instances to be enumerated such that each instance that is associated with the
1355 source instance using an association whose creation class or one of its superclasses does
1356 not have the name specified in *AssociationClassName*, is removed from the set of
1357 instances to be enumerated. There shall be no validity checking performed for the
1358 *AssociationClassName* operation input parameter; if the specified class does not exist, the
1359 operation shall succeed with an empty result (because the filter did not match).
 - 1360 NOTE Specifying a non-NULL value for *AssociatedClassName* ensures that the returned instances
1361 have the class specified in *AssociatedClassName* as a common superclass.
 - 1362 • If the *AssociatedClassName* operation input parameter is not NULL, it acts as a restricting
1363 filter on the instances to be enumerated such that each instance whose creation class or
1364 one of its superclasses does not have the name specified in *AssociatedClassName*, is
1365 removed from the set of instances to be enumerated. There shall be no validity checking
1366 performed for the *AssociatedClassName* operation input parameter; if the specified class
1367 does not exist, the operation shall succeed with an empty result (because the filter did not
1368 match).
 - 1369 • If the *SourceRoleName* operation input parameter is not NULL, it acts as a restricting filter
1370 on the instances to be enumerated such that each instance that is associated with the
1371 source instance using an association class that has a role name on the source end that is
1372 not the role name specified in *SourceRoleName*, is removed from the set of instances to
1373 be enumerated. There shall be no validity checking performed for the *SourceRoleName*
1374 operation input parameter; if the specified role does not exist, the operation shall succeed
1375 with an empty result (because the filter did not match).
 - 1376 • If the *AssociatedRoleName* operation input parameter is not NULL, it acts as a restricting
1377 filter on the instances to be enumerated such that each instance that is associated with the
1378 source instance using an association class that has a role name on the end referencing
1379 that instance that is not the role name specified in *AssociatedRoleName*, is removed from
1380 the set of instances to be enumerated. There shall be no validity checking performed for
1381 the *AssociatedRoleName* operation input parameter; if the specified role does not exist, the
1382 operation shall succeed with an empty result (because the filter did not match).
- 1383 The consistency model defined in 5.8 applies.

1384 **Preconditions:**

- 1385
- 1386
- The namespace of the source instance referenced by *SourceInstancePath* shall exist. If it does not exist, the operation shall fail, indicating WIPG0204.
- 1387
- The namespace of any returned instance paths shall exist. If it does not exist, the operation may fail, indicating WIPG0204. Note that cross-namespace association traversals may return instances in a server or namespace that is different from the server or namespace of the source instance.
- 1388
- 1389
- 1390
- The creation class of any returned instance paths shall exist in their namespace. If it does not exist, the operation may fail, indicating WIPG0214.
- 1391
- 1392

1393 **Postconditions:**

- 1394
- The instance paths shall have been returned as described in the Description paragraph for this operation.
- 1395
- Requirements on ACID properties:
- 1396
- Atomicity: N/A
- 1397
- Update Consistency: N/A
- 1398
- Isolation: Required at the level of single instances, as defined in 5.8.
- 1399
- Durability: N/A
- 1400

1401 **Error Messages:**

1402

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM server is shutting down	Optional	Infrastructure	
WIPG0240	WBEM server limits are exceeded	Optional	Infrastructure, class implem.	
WIPG0204	Namespace not found	Mandatory	Infrastructure	For input namespace
WIPG0203	Operation not supported by WBEM server infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure, class implem.	
WIPG0228	Operation not supported by class implementation	Mandatory	Class implem.	
WIPG0204	Namespace not found	Optional	Infrastructure	For namespace of returned instance paths
WIPG0214	Class not found	Optional	Infrastructure	

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0243	Timeout	Optional	Infrastructure, class implem.	
WIPG0227	Other failure	Optional	Infrastructure, class implem.	

1403

1404 NOTE Version 1.1 of this specification removed WIPG0213 (Instance not found) from this table.

1405 **6.4.5 References (deprecated)**

1406 **Purpose:**

1407 Enumerate the association instances that reference a given source instance and return their instance
 1408 representations and instance paths.

1409 **Operation input parameters:**

1410

Generic Name	Generic Type	Requirement	Description
SourceInstancePath	InstancePath	Mandatory	Instance path of the source instance (Context Parameter)
AssociationClassName	ClassName	Mandatory	NULL, or name of the association class, acting as a restricting filter on the returned instances
SourceRoleName	PropertyName	Mandatory	NULL, or name of the role on the source end of the association, acting as a restricting filter on the returned instances
IncludeClassOrigin	boolean	Conditional	Deprecated: Indicates whether class origin information for any returned properties is to be included, as defined in 6.2.1 Condition: WBEM protocol supports client side control of returning class origin information.
IncludedProperties	PropertyName []	Optional	NULL, or unordered set of property names to be included, acting as a restricting filter on the properties included in the returned instances
ExcludeSubclassProperties	boolean	Optional	Indicates whether properties added by subclasses of the association class are to be excluded, acting as a restricting filter on the properties included in the returned instances

1411

1412 **Operation output parameters:**

1413

Generic Name	Generic Type	Requirement	Description
InstanceList	InstanceSpecificationWithPath []	Mandatory	Sequence of the returned instance representations and instance paths

1414

1415 **Description:**

1416 The *References* operation enumerates association instances that reference the source instance
1417 referenced by *SourceInstancePath* and returns their instance representations and instance paths.

1418 The *References* operation has been deprecated in version 1.1.0 of this document. Use
1419 *OpenReferences* instead (see 6.5.7).

1420 The set of association instances to be enumerated shall be determined using the following algorithm:

1421 • Initially, the set of instances to be enumerated is the set of all instances referencing the
1422 source instance specified in *SourceInstancePath*. These associations may be instances of
1423 different association classes. If the source instance does not exist, the operation shall
1424 succeed with an empty result set (even when its creation class does not exist). However, if
1425 the namespace of the source instance does not exist, the operation shall fail, indicating
1426 WIPG0204.

1427 • If the *AssociationClassName* operation input parameter is not NULL, it acts as a restricting
1428 filter on the instances to be enumerated such that each association instance whose
1429 creation class or one of its superclasses does not have the name specified in
1430 *AssociationClassName*, is removed from the set of instances to be enumerated. There
1431 shall be no validity checking performed for the *AssociationClassName* operation input
1432 parameter; if the specified class does not exist, the operation shall succeed with an empty
1433 result (because the filter did not match).

1434 NOTE Specifying a non-NULL value for *AssociationClassName* ensures that the returned
1435 instances have the class specified in *AssociationClassName* as a common superclass.

1436 • If the *SourceRoleName* operation input parameter is not NULL, it acts as a restricting filter
1437 on the instances to be enumerated such that each association instance whose creation
1438 class does not have the role name specified in *SourceRoleName* on the end referencing
1439 the source instance, is removed from the set of instances to be enumerated. There shall be
1440 no validity checking performed for the *SourceRoleName* operation input parameter; if the
1441 specified role does not exist, the operation shall succeed with an empty result (because the
1442 filter did not match).

1443 NOTE Version 1.1 of this specification removed the *AssociatedClassName* and *AssociatedRoleName* filters
1444 from this operation.

1445 The consistency model defined in 5.8 applies.

1446 The set of properties to be included in each returned instance representation shall be determined
1447 using the following algorithm:

1448 • Initially, the set of properties to be included is the set of properties exposed by the creation
1449 class of the instance. This includes all the duplicates of any duplicate non-overridden
1450 properties.

1451 • If the *IncludedProperties* operation input parameter is supported by the WBEM protocol
1452 and if its value is not NULL, it acts as a restricting filter on the properties to be included in
1453 the returned instance representations such that any properties exposed by the creation
1454 class of the instance that are not named in that operation parameter are removed from the
1455 set of properties to be included. Any duplicate or invalid property names in the
1456 *IncludedProperties* operation input parameter shall be ignored. A non-NULL empty
1457 *IncludedProperties* list removes all properties from the set of properties to be included

- 1458 • If the *ExcludeSubclassProperties* operation input parameter is supported by the WBEM
1459 protocol and if its value is TRUE, it acts as a restricting filter on the properties to be
1460 included in the returned instance representations such that any properties not exposed by
1461 the class specified in *AssociationClassName* are removed from the set of properties to be
1462 included.
- 1463 • Conformant WBEM protocols may specify rules that cause properties with a value of NULL
1464 to be removed from the set of properties to be included.

1465 **Preconditions:**

- 1466 • The namespace of the source instance referenced by *SourceInstancePath* shall exist. If it does
1467 not exist, the operation shall fail, indicating WIPG0204.
- 1468 • The *IncludedProperties* operation parameter, if supported by the WBEM protocol, shall only be
1469 specified with a non-NULL value if the *AssociationClassName* operation input parameter is also
1470 non-NULL. If this is not satisfied, the operation shall fail, indicating WIPG0208.
- 1471 • The *ExcludeSubclassProperties* operation parameter, if supported by the WBEM protocol, shall
1472 only be specified with a TRUE value if the *AssociationClassName* operation input parameter is
1473 non-NULL. If this is not satisfied, the operation shall fail, indicating WIPG0208.
- 1474 • The namespace of any returned instance paths shall exist. If it does not exist, the operation may
1475 fail, indicating WIPG0204. Note that cross-namespace association traversals may return
1476 instance paths in a server or namespace that is different from the server or namespace of the
1477 source instance.
- 1478 • The creation class of any returned instance paths shall exist in their namespace. If it does not
1479 exist, the operation may fail, indicating WIPG0214.

1480 **Postconditions:**

- 1481 • The instance representations and instance paths shall have been returned as described in the
1482 Description paragraph for this operation.
- 1483 • Requirements on ACID properties:
1484 – Atomicity: N/A
1485 – Update Consistency: N/A
1486 – Isolation: Required at the level of single instances, as defined in 5.8.
1487 – Durability: N/A

1488 **Error Messages:**

1489

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM server is shutting down	Optional	Infrastructure	
WIPG0240	WBEM server limits are exceeded	Optional	Infrastructure, class implem.	
WIPG0204	Namespace not found	Mandatory	Infrastructure	For input namespace
WIPG0203	Operation not supported by WBEM server infrastructure	Mandatory	Infrastructure	

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure, class implem.	
WIPG0228	Operation not supported by class implementation	Mandatory	Class implem.	
WIPG0204	Namespace not found	Optional	Infrastructure	For namespace of returned instance paths
WIPG0214	Class not found	Optional	Infrastructure	
WIPG0243	Timeout	Optional	Infrastructure, class implem.	
WIPG0227	Other failure	Optional	Infrastructure, class implem.	

1490

1491 NOTE Version 1.1 of this specification removed WIPG0213 (Instance not found) from this table.

1492 **6.4.6 ReferenceNames (deprecated)**1493 **Purpose:**

1494 Enumerate the association instances that reference a given source instance and return their instance
1495 paths.

1496 **Operation input parameters:**

1497

Generic Name	Generic Type	Requirement	Description
SourceInstancePath	InstancePath	Mandatory	Instance path of the source instance (Context Parameter)
AssociationClassName	ClassName	Mandatory	NULL, or name of the association class, acting as a restricting filter on the returned instances
SourceRoleName	PropertyName	Mandatory	NULL, or name of the role on the source end of the association, acting as a restricting filter on the returned instances

1498

1499 **Operation output parameters:**

1500

Generic Name	Generic Type	Requirement	Description
InstancePathList	InstancePath []	Mandatory	Sequence of the returned instance paths

1501

1502 **Description:**

1503 The *ReferenceNames* operation enumerates the association instances that reference the source
 1504 instance referenced by *SourceInstancePath* and returns their instance paths.

1505 The *ReferenceNames* operation has been deprecated in version 1.1.0 of this document. Use
 1506 *OpenReferences* instead (see 6.5.7).

1507 The set of association instances to be enumerated shall be determined using the following algorithm:

1508 • Initially, the set of instances to be enumerated is the set of all association instances
 1509 referencing the source instance referenced by *SourceInstancePath*. These associations
 1510 may be instances of different association classes. If the source instance does not exist, the
 1511 operation shall succeed with an empty result set (even when its creation class does not
 1512 exist). However, if the namespace of the source instance does not exist, the operation shall
 1513 fail, indicating WIPG0204.

1514 • If the *AssociationClassName* operation input parameter is not NULL, it acts as a restricting
 1515 filter on the instances to be enumerated such that each association instance whose
 1516 creation class or one of its superclasses does not have the name specified in
 1517 *AssociationClassName*, is removed from the set of instances to be enumerated. There
 1518 shall be no validity checking performed for the *AssociationClassName* operation input
 1519 parameter; if the specified class does not exist, the operation shall succeed with an empty
 1520 result (because the filter did not match).

1521 NOTE Specifying a non-NULL value for *AssociationClassName* ensures that the returned
 1522 instances have the class specified in *AssociationClassName* as a common superclass.

1523 • If the *SourceRoleName* operation input parameter is not NULL, it acts as a restricting filter
 1524 on the instances to be enumerated such that each association instance whose creation
 1525 class does not have the role name specified in *SourceRoleName* on the end referencing
 1526 the source instance, is removed from the set of instances to be enumerated. There shall be
 1527 no validity checking performed for the *SourceRoleName* operation input parameter; if the
 1528 specified role does not exist, the operation shall succeed with an empty result (because the
 1529 filter did not match).

1530 NOTE Version 1.1 of this specification removed the *AssociatedClassName* and *AssociatedRoleName* filters
 1531 from this operation.

1532 The consistency model defined in 5.8 applies.

1533 **Preconditions:**

- 1534 • The namespace of the source instance referenced by *SourceInstancePath* shall exist. If it does
 1535 not exist, the operation shall fail, indicating WIPG0204.
- 1536 • The namespace of any returned instance paths shall exist. If it does not exist, the operation may
 1537 fail, indicating WIPG0204. Note that cross-namespace association traversals may return
 1538 instance paths in a server or namespace that is different from the server or namespace of the
 1539 source instance.
- 1540 • The creation class of any returned instance paths shall exist in their namespace. If it does not
 1541 exist, the operation may fail, indicating WIPG0214.

1542 **Postconditions:**

- 1543 • The instance paths shall have been returned as described in the Description paragraph for this
 1544 operation.
- 1545 • Requirements on ACID properties:
 1546 – Atomicity: N/A

- 1547 – Update Consistency: N/A
- 1548 – Isolation: Required at the level of single instances, as defined in 5.8.
- 1549 – Durability: N/A

1550 **Error Messages:**

1551

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM server is shutting down	Optional	Infrastructure	
WIPG0240	WBEM server limits are exceeded	Optional	Infrastructure, class implem.	
WIPG0204	Namespace not found	Mandatory	Infrastructure	For input namespace
WIPG0203	Operation not supported by WBEM server infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure, class implem.	
WIPG0228	Operation not supported by class implementation	Mandatory	Class implem.	
WIPG0204	Namespace not found	Optional	Infrastructure	For namespace of returned instance paths
WIPG0214	Class not found	Optional	Infrastructure	
WIPG0243	Timeout	Optional	Infrastructure, class implem.	
WIPG0227	Other failure	Optional	Infrastructure, class implem.	

1552

1553 NOTE Version 1.1 of this specification removed WIPG0213 (Instance not found) from this table.

1554 **6.5 Pulled instance enumeration operations**

1555 This subclause defines server operations that enumerate instances and return their representations and
1556 instance paths by means of subsequent pull operations.

1557 The common pattern for these operations is that an enumeration session gets established through an
1558 "Open" operation, also establishing the kind of operation and the kind of items to be returned (instance
1559 representations together with instance paths, or just instance paths), and subsequent repeated
1560 executions of a "Pull" operation on the enumeration session are used to retrieve the items. Optionally, the
1561 "Open" operation can also pull a first set of items.

- 1562 The pulled instance enumeration operations consist of the following individual operations:
- 1563 • Open operations:
 - 1564 OpenEnumerateInstances – Open an enumeration of instances of a given class for returning
 - 1565 their representations and instance paths
 - 1566 OpenEnumerateInstancePaths (deprecated) – Open an enumeration of instances of a given
 - 1567 class for returning only their instance paths
 - 1568 OpenAssociators – Open an enumeration of instances associated to a given source instance for
 - 1569 returning their representations and instance paths
 - 1570 OpenAssociatorPaths (deprecated) – Open an enumeration of instances associated to a given
 - 1571 source instance for returning only their instance paths
 - 1572 OpenReferences – Open an enumeration of association instances referencing a given source
 - 1573 instance for returning their representations and instance paths
 - 1574 OpenReferencePaths (deprecated) – Open an enumeration of association instances
 - 1575 referencing a given source instance for returning only their instance paths
 - 1576 OpenQueryInstances – Open an enumeration of instances representing a query result for
 - 1577 returning only their instance representations
 - 1578 • Pull operations:
 - 1579 PullInstancesWithPath – Pull operation for retrieving instance representations with instance
 - 1580 paths
 - 1581 PullInstancePaths (deprecated) – Pull operation for retrieving instance paths
 - 1582 PullInstances – Pull operation for retrieving instance representations (without instance paths),
 - 1583 representing query results
 - 1584 • Other operations:
 - 1585 CloseEnumeration – Close an open enumeration
 - 1586 EnumerationCount (deprecated) – Estimate number of remaining items in an open enumeration

1587 **6.5.1 General behavioral rules**

1588 A central concept of the pulled instance enumeration operations is the "enumeration session". An
 1589 enumeration session can be thought of as a context in which the operations perform their work, and
 1590 which determines the set of instances to be enumerated. In order to process the operations related to an
 1591 enumeration session, some of the operation parameters of the Open operation need to be maintained as
 1592 long as the enumeration session is open, as well as some state data about where the enumeration
 1593 session is with respect to instances already returned.

1594 From a WBEM client's perspective, an enumeration session is represented as an enumeration context
 1595 value. A successful Open operation establishes the enumeration session and returns an enumeration
 1596 context value representing the open enumeration session. The enumeration context value is used as an
 1597 operation input/output parameter in subsequent Pull operations on that enumeration session. The
 1598 enumeration context value shall uniquely identify the open enumeration session within the target
 1599 namespace of the Open operation that established the enumeration session. This does not require the
 1600 enumeration context value to be time-unique, i.e., it may be reused for a new enumeration session after
 1601 the old enumeration session was closed. It is valid for a WBEM server to use NULL as an enumeration
 1602 context value representing a closed enumeration session, but a WBEM client shall not rely on that to
 1603 detect that an enumeration session has been closed.

1604 Defining the enumeration context value in Pull operations not only as an operation input parameter but
1605 also as an operation output parameter allows the WBEM server to change the enumeration context value
1606 during the execution of a Pull operation. This allows for different implementation approaches for the
1607 WBEM server, which are transparent for the WBEM client.

1608 Example approaches are:

- 1609 • maintaining any state data describing the enumeration session internally in the WBEM server.
1610 In this approach, the enumeration context value does not need to change in subsequent Pull
1611 operations. It is used by the WBEM server only to identify the internal state data for the open
1612 enumeration session, but it is not used to store any of the state data in it. A variation of this
1613 approach is to hand back modified enumeration context values for additional WBEM server side
1614 sequence checking.
- 1615 • maintaining any state data describing the enumeration session on the WBEM client side only. In
1616 this approach, all state data is stored in the enumeration context value, and the WBEM server
1617 does not maintain any state data about the enumeration session, essentially being completely
1618 stateless with respect to the enumeration session.
- 1619 • a combination of the two previous approaches

1620 A WBEM server may support keeping enumeration sessions open across connection terminations and
1621 shutdowns of the server. Objects may be created, deleted or modified concurrently with an enumeration
1622 session that involves these objects. Such changes may or may not be reflected in the enumeration set.
1623 Therefore, there is no guarantee to the WBEM client that the enumeration set represents a consistent
1624 snapshot of its objects at a point in time. However, the WBEM server should make a best effort attempt
1625 for the returned enumeration set to represent a consistent snapshot of its objects at a point in time. The
1626 order of objects in the enumeration set is undefined.

1627 This specification does not define any restrictions on the number of enumeration sessions that can be
1628 established or executed on concurrently in the same WBEM server or by the same WBEM client. This
1629 remains true even if the enumeration sets of such concurrently established enumeration sessions contain
1630 the same objects.

1631 With the exception of the CloseEnumeration operation, all operations on a particular enumeration session
1632 shall be executed sequentially. An enumeration session can be open or closed. The enumeration session
1633 is considered open if operations using its enumeration context value as an operation input parameter can
1634 be executed successfully. It is opened by the successful completion of an Open operation and closed by
1635 one of the following:

- 1636 • Successful completion of a CloseEnumeration operation
- 1637 • Successful completion of an Open or Pull operation that has its *EndOfSequence* operation
1638 output parameter set to TRUE. In other words, reaching the end of the enumeration set closes
1639 the enumeration session implicitly
- 1640 • Unsuccessful completion of a Pull operation when *ContinueOnError* had not been requested
- 1641 • WBEM server side decision to close the enumeration session based upon an operation timeout
- 1642 • WBEM server side decision to close an enumeration session during an operation on that
1643 enumeration session based upon exceeding server limits

1644 A conformant WBEM server may support closure of enumeration sessions based upon exceeding server
1645 limits. Potential examples for such a decision may be Pull operations with no objects requested that are
1646 repeated with a high frequency on the same enumeration session. If a WBEM server supports closure of
1647 enumeration sessions based upon exceeding server limits, it shall make the decision to close an
1648 enumeration session during an operation on that enumeration session. (There is no way to indicate the
1649 reason for the closure if the decision is made elsewhere.)

1650 6.5.2 Common operation parameters for the open operations

1651 This subclause defines commonly used operation parameters for the Open operations. The description of
1652 the individual Open operations references these operation parameters as appropriate. However, not
1653 every Open operation uses every one of these common operation parameters.

1654 6.5.2.1 EnumerationContext

1655 The *EnumerationContext* operation output parameter is the enumeration context value representing the
1656 enumeration session. See 6.5.1 for a definition of the concepts of *enumeration session* and *enumeration*
1657 *context value*.

1658 6.5.2.2 EndOfSequence

1659 NOTE This operation output parameter is also used for Pull operations.

1660 The *EndOfSequence* operation output parameter indicates whether the enumeration session is
1661 exhausted.

1662 If *EndOfSequence* is TRUE upon successful completion of an operation, no more objects are available
1663 and the WBEM server shall have closed the enumeration session, releasing any possibly allocated
1664 compute resources related to the enumeration session.

1665 If the returned enumeration set is empty, it is valid for a WBEM server to set *EndOfSequence* to TRUE,
1666 even if *MaxObjectCount* was 0. In this case, the enumeration session will be closed upon successful
1667 completion of the operation.

1668 If *EndOfSequence* is FALSE upon successful completion of an operation, there may be additional
1669 elements available and the WBEM server shall not have closed the enumeration session.

1670 6.5.2.3 FilterQueryLanguage and FilterQueryString

1671 The *FilterQueryLanguage* and *FilterQueryString* operation input parameters define a filter query that acts
1672 as an additional restricting filter on the set of instances about which information is returned.

1673 Support for the *FilterQueryLanguage* and *FilterQueryString* operation parameters is conditional on
1674 support in the WBEM protocol for filter queries in pulled instance enumeration operations.

1675 If the WBEM protocol supports filter queries in pulled instance enumeration operations, the following rules
1676 apply:

- 1677 • Conformant WBEM protocols shall require that the DMTF Filter Query Language (FQL) defined
1678 in [DSP0212](#) is supported for the filter queries. Conformant WBEM protocols may support
1679 additional filter query languages.
- 1680 • If *FilterQueryLanguage* is not NULL, additional filtering is requested and the following rules
1681 apply:
 - 1682 – *FilterQueryLanguage* shall specify a valid query language and *FilterQueryString* shall
1683 be a valid query in that query language. Neither the query language nor the format of
1684 the filter query is defined by this specification. Conformant WBEM protocols shall
1685 define a mechanism whereby WBEM servers can declare the set of query languages
1686 that are valid for *FilterQueryLanguage*.
 - 1687 – A filter query may specify any result set (e.g., SELECT list), but because the purpose
1688 of the filter query is to restrict the set of instances about which information is returned,
1689 its result set shall be ignored. The filter query shall not define any ordering criteria.
1690 The filter query shall not define any grouping of objects. Operations using filter queries
1691 may specify additional constraints on the filter query.

- 1692 – If the WBEM server infrastructure does not support filtered enumerations, the WBEM
 1693 server shall return failure with message WIPG0237 (Filter queries not supported by
 1694 WBEM server infrastructure).
- 1695 – If the CIM class implementation does not support filtered enumerations, the WBEM
 1696 server shall return failure with message WIPG0244 (Filter queries not supported by
 1697 class implementation).
- 1698 • If *FilterQueryLanguage* is NULL, no additional filtering shall take place, and *FilterQueryString*
 1699 shall be NULL.
- 1700 – If *FilterQueryString* is not NULL, the WBEM server shall return failure with message
 1701 WIPG0208 (Invalid operation input parameter value).

1702 If the WBEM protocol does not support filter queries in pulled instance enumeration operations, no
 1703 additional filtering shall take place.

1704 6.5.2.4 OperationTimeout

1705 The *OperationTimeout* operation input parameter determines the "operation timeout". The operation
 1706 timeout is the minimum time the WBEM server shall maintain the open enumeration session after the last
 1707 Open or Pull operation (unless the enumeration session was closed during that last operation). If the
 1708 operation timeout is exceeded, the WBEM server may close the enumeration session at any time,
 1709 releasing any possibly allocated compute resources related to the enumeration session.

1710 Support for the *OperationTimeout* operation parameter in a conformant WBEM protocol is mandatory.

1711 An *OperationTimeout* of 0 means that there is no operation timeout, i.e., the enumeration session is never
 1712 closed based on time.

1713 If *OperationTimeout* is NULL, the WBEM server shall choose an operation timeout.

1714 All other values for *OperationTimeout* specify the operation timeout in seconds.

1715 A WBEM server may restrict the set of allowable values for *OperationTimeout*. This specifically includes
 1716 the possibility for the WBEM server to not allow 0 (no timeout). If the specified value is not an allowable
 1717 value, the WBEM server shall return failure with error message WIPG0242 (Invalid timeout). Conformant
 1718 WBEM protocols shall define a mechanism whereby WBEM servers can declare the allowable values for
 1719 *OperationTimeout*.

1720 6.5.2.5 ContinueOnError

1721 The *ContinueOnError* operation input parameter, if TRUE, requests continuation on error. Continuation on
 1722 error is the ability to resume an enumeration session successfully after a Pull operation that returned an
 1723 error. A conformant WBEM server may support continuation on error. Conformant WBEM protocols shall
 1724 define a mechanism whereby WBEM servers can declare support for continuation on error.

1725 Support for the *ContinueOnError* operation parameter is conditional on support in the WBEM protocol for
 1726 client side control of continuation on error for pulled instance enumeration operations.

1727 If the WBEM protocol supports client side control of continuation on error for pulled instance enumeration
 1728 operations, the following rules apply:

- 1729 • If a WBEM server does not support continuation on error and if *ContinueOnError* is TRUE, it
 1730 shall return failure with error message WIPG0235 (Continuation on error not supported).
- 1731 • If a WBEM server supports continuation on error, it shall support it as follows: If
 1732 *ContinueOnError* is TRUE, the enumeration session shall remain open when a Pull operation
 1733 returns failure, and any subsequent successful Pull operations shall return the set of elements
 1734 that would have been returned if the failing Pull operations had been successful, subject to the

1735 consistency rules defined in 5.8. If *ContinueOnError* is FALSE, the enumeration session shall
 1736 be closed when a Pull operation returns failure.

1737 If the WBEM protocol does not support client side control of continuation on error for pulled instance
 1738 enumeration operations, it shall define requirements for the behavior of the WBEM server with respect to
 1739 continuation on error.

1740 **6.5.2.6 MaxObjectCount**

1741 NOTE This operation output parameter is also used for Pull operations.

1742 The *MaxObjectCount* operation input parameter defines the maximum number of objects that may be
 1743 returned by this operation. Any uint32 number is valid, including 0. The WBEM server may deliver any
 1744 number of objects up to *MaxObjectCount* but shall not deliver more than *MaxObjectCount* objects.

1745 Support for the *MaxObjectCount* operation parameter in a conformant WBEM protocol is mandatory.

1746 A conformant WBEM server implementation may choose to never return any elements during an
 1747 operation, regardless of the value of *MaxObjectCount*.

1748 A WBEM client may use a *MaxObjectCount* value of 0 to specify that it does not want to retrieve any
 1749 instances in the operation.

1750 **6.5.3 OpenEnumerateInstances**

1751 **Purpose:**

1752 Establish and open an enumeration session for enumerating the instances of a given class and
 1753 optionally return a first set of their instance representations and instance paths.

1754 **Operation input parameters:**

1755

Generic Name	Generic Type	Requirement	Description
EnumClassPath	ClassPath	Mandatory	Class path of the class whose instances are to be enumerated (Context Parameter)
FilterQueryString	QueryString	Conditional	NULL, or query string of a filter query that is acting as an additional restricting filter on the set of instances to be enumerated, as defined in 6.5.2.3 Condition: WBEM protocol supports filter queries for pulled instance enumeration operations.
FilterQueryLanguage	QueryLanguage	Conditional	NULL, or query language of the filter query specified in <i>FilterQueryString</i> , as defined in 6.5.2.3 Condition: WBEM protocol supports filter queries for pulled instance enumeration operations.
IncludeClassOrigin	boolean	Conditional	Deprecated: Indicates whether class origin information for any returned properties is to be included, as defined in 6.2.1 Condition: WBEM protocol supports client side control of returning class origin information.

Generic Name	Generic Type	Requirement	Description
IncludedProperties	PropertyName []	Optional	NULL, or unordered set of property names to be included, acting as a restricting filter on the properties included in the returned instance representations
ExcludeSubclassProperties	boolean	Optional	Indicates whether properties added by subclasses of the class used for the enumeration are to be excluded, acting as a restricting filter on the properties included in the returned instance representations
OperationTimeout	uint32	Mandatory	Operation timeout, as defined in 6.5.2.4
ContinueOnError	boolean	Conditional	Indicates whether the enumeration session should be continued in case of error, as defined in 6.5.2.5 Condition: WBEM protocol supports client side control of continuation on error for pulled instance enumeration operations.
MaxObjectCount	uint32	Mandatory	Maximum number of instances that may be returned by this operation, as defined in 6.5.2.6

1756

1757 **Operation output parameters:**

1758

Generic Name	Generic Type	Requirement	Description
InstanceList	InstanceSpecificationWithPath []	Mandatory	Sequence of the returned first set of instance representations and instance paths
EnumerationContext	EnumerationContext	Mandatory	Enumeration context value, as defined in 6.5.2.1
EndOfSequence	boolean	Mandatory	Indicates end of sequence for the enumeration session, as defined in 6.5.2.2

1759

1760 **Description:**

1761 The *OpenEnumerateInstances* operation establishes and opens an enumeration session for
 1762 enumerating all instances of the class referenced by *EnumClassPath*, including instances of any of
 1763 its subclasses. That enumeration session allows retrieving the instance representations and instance
 1764 paths of these instances through successive *PullInstancesWithPath* operations (see 6.5.11).
 1765 Retrieval of a first set of instance representations and instance paths may be requested by setting
 1766 *MaxObjectCount* to a value > 0.

1767 The set of instances to be enumerated throughout the entire enumeration session shall be
 1768 determined using the following algorithm:

- 1769 • Initially, the set of instances to be enumerated is the set of instances in the namespace of
 1770 the class referenced by *EnumClassPath*, whose creation class is the class referenced by
 1771 *EnumClassPath* or a subclass of that class.
- 1772 • If the WBEM protocol supports filter queries for pulled instance enumeration operations
 1773 (that is, the *FilterQueryString* and *FilterQueryLanguage* operation parameters) and
 1774 *FilterQueryLanguage* is not NULL, *FilterQueryString* acts as a restricting filter on the

- 1775 instances to be enumerated such that any instances not selected by the filter query for its
 1776 result set are removed from the set of instances. The filter query shall query only the class
 1777 referenced by *EnumClassPath*. See also 6.5.2.3.
- 1778 The set of instances to be enumerated throughout the entire enumeration session should not contain
 1779 any duplicate instances, as defined in 5.8.4. Because instances to be enumerated all exist in the
 1780 same namespace, a determination of duplicate instances (for example by a WBEM client) can be
 1781 done on the basis of their model paths only.
- 1782 The set of instances to be returned (as instance representations and instance paths) is the first set of
 1783 instances from the set of instances to be enumerated throughout the entire enumeration session,
 1784 such that no more than *MaxObjectCount* instances are returned. Returning no instances does not
 1785 imply that the enumeration session has been exhausted. Only the *EndOfSequence* operation output
 1786 parameter indicates whether the enumeration session has been exhausted.
- 1787 The set of properties to be included in any returned instance representations shall be determined
 1788 using the following algorithm:
- 1789 • Initially, the set of properties to be included is the set of properties exposed by the creation
 1790 class of the instance. This includes all the duplicates of any duplicate non-overridden
 1791 properties.
 - 1792 • If the *IncludedProperties* operation input parameter is supported by the WBEM protocol
 1793 and if its value is not NULL, it acts as a restricting filter on the properties to be included in
 1794 the returned instance representations such that any properties exposed by the creation
 1795 class of the instance that are not named in that operation parameter are removed from the
 1796 set of properties to be included. Any duplicate or invalid property names in the
 1797 *IncludedProperties* operation input parameter shall be ignored. A non-NULL empty
 1798 *IncludedProperties* list removes all properties from the set of properties to be included.
 - 1799 • If the *ExcludeSubclassProperties* operation input parameter is supported by the WBEM
 1800 protocol and if its value is TRUE, it acts as a restricting filter on the properties to be
 1801 included in the returned instance representations such that any properties not exposed by
 1802 the class referenced by *EnumClassPath* are removed from the set of properties to be
 1803 included. In other words, the set of properties is restricted to the properties exposed by the
 1804 enumeration class.
 - 1805 • Conformant WBEM protocols may specify rules that cause properties with a value of NULL
 1806 to be removed from the set of properties to be included.

1807 **Preconditions:**

- 1808 • The class referenced by *EnumClassPath* shall exist. If it does not exist, the operation shall fail,
 1809 indicating WIPG0214.
- 1810 • The namespace of the class referenced by *EnumClassPath* shall exist. If it does not exist, the
 1811 operation shall fail, indicating WIPG0204.
- 1812 • If a filter query is specified,
 - 1813 – the query language specified in the *FilterQueryLanguage* operation parameter shall be
 1814 valid. If this is not satisfied, the operation shall fail, indicating WIPG0221.
 - 1815 – the query specified in the *FilterQueryString* operation parameter shall be a valid query in
 1816 the query language specified in the *FilterQueryLanguage* operation parameter. If this is not
 1817 satisfied, the operation shall fail, indicating WIPG0222 or WIPG0223.

1818 **Postconditions:**

- 1819
- The enumeration session shall have been established and opened.
- 1820
- A first set of instance representations and instance paths shall have been returned as described in the Description paragraph for this operation.
- 1821
- Requirements on ACID properties:
- 1822
- Atomicity: Required (related to the creation of an enumeration context that is maintained by the WBEM server)
- 1823
- Update Consistency: N/A
- 1824
- Isolation: Required at the level of single instances, as defined in 5.8.
- 1825
- Durability: Required (related to creation of an enumeration context that is maintained by the WBEM server)
- 1826
- Durability: Required (related to creation of an enumeration context that is maintained by the WBEM server)
- 1827
- Durability: Required (related to creation of an enumeration context that is maintained by the WBEM server)
- 1828

1829 **Error Messages:**

1830

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM server is shutting down	Optional	Infrastructure	
WIPG0240	WBEM server limits are exceeded	Optional	Infrastructure, class implem.	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM server infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0242	Invalid timeout	Mandatory	Infrastructure, class implem.	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure, class implem.	
WIPG0235	Continuation on error not supported	Mandatory	Infrastructure, class implem.	
WIPG0237	Filter queries not supported by WBEM server infrastructure	Optional	Infrastructure	
WIPG0244	Filter queries not supported by class implementation	Optional	Class implem.	
WIPG0221	Unknown query language	Mandatory	Infrastructure, class implem.	
WIPG0222	Query language feature not supported	Mandatory	Infrastructure, class implem.	

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0223	Invalid query	Mandatory	Infrastructure, class implem.	
WIPG0214	Class not found	Mandatory	Infrastructure	
WIPG0228	Operation not supported by class implementation	Mandatory	Class implem.	
WIPG0243	Timeout	Optional	Infrastructure, class implem.	
WIPG0227	Other failure	Optional	Infrastructure, class implem.	

1831

1832 **6.5.4 OpenEnumerateInstancePaths (deprecated)**

1833 **Purpose:**

1834 Establish and open an enumeration session for enumerating the instances of a class and optionally
 1835 return a first set of their instance paths.

1836 **Operation input parameters:**

1837

Generic Name	Generic Type	Requirement	Description
EnumClassPath	ClassPath	Mandatory	Class path of the class whose instances are to be enumerated (Context Parameter)
FilterQueryString	QueryString	Conditional	NULL, or query string of a filter query that is acting as an additional restricting filter on the set of enumerated instances, as defined in 6.5.2.3 Condition: WBEM protocol supports filter queries for pulled instance enumeration operations.
FilterQueryLanguage	QueryLanguage	Conditional	NULL, or query language of the filter query specified in <i>FilterQueryString</i> , as defined in 6.5.2.3. Condition: WBEM protocol supports filter queries for pulled instance enumeration operations.
OperationTimeout	uint32	Mandatory	Operation timeout, as defined in 6.5.2.4
ContinueOnError	boolean	Conditional	Indicates whether the enumeration session should be continued in case of error, as defined in 6.5.2.5 Condition: WBEM protocol supports client side control of continuation on error for pulled instance enumeration operations.
MaxObjectCount	uint32	Mandatory	Maximum number of instance paths that may be returned by this operation, as defined in 6.5.2.6

1838

1839 **Operation output parameters:**

1840

Generic Name	Generic Type	Requirement	Description
InstancePathList	InstancePath []	Mandatory	Sequence of the returned first set of instance paths
EnumerationContext	EnumerationContext	Mandatory	Enumeration context value, as defined in 6.5.2.1
EndOfSequence	boolean	Mandatory	Indicates end of sequence for the enumeration session, as defined in 6.5.2.2

1841

1842 **Description:**

1843 The *OpenEnumerateInstancePaths* operation establishes and opens an enumeration session for
 1844 enumerating the instance paths of all instances of the class referenced by *EnumClassPath*, including
 1845 of instances of any of its subclasses. That enumeration session allows retrieving the instance paths
 1846 of these instances through successive *PullInstancePaths* operations (see 6.5.12). Retrieval of a first
 1847 set of those instance paths may be requested by setting *MaxObjectCount* to a value > 0.

1848 The *OpenEnumerateInstancePaths* operation has been deprecated in version 1.1.0 of this
 1849 document. Use *OpenEnumerateInstances* instead (see 6.5.3).

1850 The set of instances to be enumerated throughout the entire enumeration session shall be
 1851 determined using the following algorithm:

- 1852 • Initially, the set of instances to be enumerated is the set of instances in the namespace of
 1853 the class referenced by *EnumClassPath*, whose creation class is the class referenced by
 1854 *EnumClassPath* or a subclass of that class.
- 1855 • If the WBEM protocol supports filter queries for pulled instance enumeration operations
 1856 (that is, the *FilterQueryString* and *FilterQueryLanguage* operation parameters) and
 1857 *FilterQueryLanguage* is not NULL, *FilterQueryString* acts as a restricting filter on the
 1858 instances to be enumerated such that any instances not selected by the filter query for its
 1859 result set are removed from the set of instances. The filter query shall query only the class
 1860 referenced by *EnumClassPath*. See also 6.5.2.3.

1861 The set of instances to be enumerated throughout the entire enumeration session should not contain
 1862 any duplicate instances, as defined in 5.8.4. Because the instances to be enumerated all exist in the
 1863 same namespace, a determination of duplicate instances can be done on the basis of their model
 1864 paths only.

1865 The set of instances to be returned (as instance paths) is the first set of instances from the set of
 1866 instances to be enumerated throughout the entire enumeration session, such that no more than
 1867 *MaxObjectCount* instance paths are returned. Returning no instances does not imply that the
 1868 enumeration session has been exhausted. Only the *EndOfSequence* operation output parameter
 1869 indicates whether the enumeration session has been exhausted.

1870 **Preconditions:**

- 1871 • The class referenced by *EnumClassPath* shall exist. If it does not exist, the operation shall fail,
 1872 indicating WIPG0214.
- 1873 • The namespace of the class referenced by *EnumClassPath* shall exist. If it does not exist, the
 1874 operation shall fail, indicating WIPG0204.

- 1875 • If a filter query is specified,
- 1876 – the query language specified in the *FilterQueryLanguage* operation parameter shall be
- 1877 valid. If this is not satisfied, the operation shall fail, indicating WIPG0221.
- 1878 – the query specified in the *FilterQueryString* operation parameter shall be a valid query in
- 1879 the query language specified in the *FilterQueryLanguage* operation parameter. If this is not
- 1880 satisfied, the operation shall fail, indicating WIPG0222 or WIPG0223.

1881 **Postconditions:**

- 1882 • The enumeration session shall have been established and opened.
- 1883 • A first set of instance paths shall have been returned as described in the Description paragraph
- 1884 for this operation.
- 1885 • Requirements on ACID properties:
- 1886 – Atomicity: Required (related to the creation of an enumeration context that is maintained by
- 1887 the WBEM server)
- 1888 – Update Consistency: N/A
- 1889 – Isolation: Required at the level of single instances, as defined in 5.8.
- 1890 – Durability: Required (related to creation of an enumeration context that is maintained by
- 1891 the WBEM server)

1892 **Error Messages:**

1893

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM server is shutting down	Optional	Infrastructure	
WIPG0240	WBEM server limits are exceeded	Optional	Infrastructure, class implem.	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM server infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0242	Invalid timeout	Mandatory	Infrastructure, class implem.	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure, class implem.	
WIPG0235	Continuation on error not supported	Mandatory	Infrastructure, class implem.	
WIPG0237	Filter queries not supported by WBEM server infrastructure	Optional	Infrastructure	

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0244	Filter queries not supported by class implementation	Optional	Class implem.	
WIPG0221	Unknown query language	Mandatory	Infrastructure, class implem.	
WIPG0222	Query language feature not supported	Mandatory	Infrastructure, class implem.	
WIPG0223	Invalid query	Mandatory	Infrastructure, class implem.	
WIPG0214	Class not found	Mandatory	Infrastructure	
WIPG0228	Operation not supported by class implementation	Mandatory	Class implem.	
WIPG0243	Timeout	Optional	Infrastructure, class implem.	
WIPG0227	Other failure	Optional	Infrastructure, class implem.	

1894

1895 **6.5.5 OpenAssociators**1896 **Purpose:**

1897 Establish and open an enumeration session for enumerating the instances that are associated with a
 1898 given source instance and optionally return a first set of their instance representations and instance
 1899 paths.

1900 **Operation input parameters:**

1901

Generic Name	Generic Type	Requirement	Description
SourceInstancePath	InstancePath	Mandatory	Instance path of the source instance (Context Parameter)
AssociationClassName	ClassName	Mandatory	NULL, or name of the association class, acting as a restricting filter on the returned instances
AssociatedClassName	ClassName	Mandatory	NULL, or name of the associated class on any far end of the association, acting as a restricting filter on the returned instances
SourceRoleName	PropertyName	Mandatory	NULL, or name of the role on the source end of the association, acting as a restricting filter on the returned instances
AssociatedRoleName	PropertyName	Mandatory	NULL, or name of the role on any far end of the association, acting as a restricting filter on the returned instances

Generic Name	Generic Type	Requirement	Description
FilterQueryString	QueryString	Conditional	NULL, or query string of a filter query that is acting as an additional restricting filter on the set of returned instances, as defined in 6.5.2.3 Condition: WBEM protocol supports filter queries for pulled instance enumeration operations.
FilterQueryLanguage	QueryLanguage	Conditional	NULL, or query language of the filter query specified in <i>FilterQueryString</i> , as defined in 6.5.2.3 Condition: WBEM protocol supports filter queries for pulled instance enumeration operations.
IncludeClassOrigin	boolean	Conditional	Deprecated: Indicates whether class origin information for any returned properties is to be included, as defined in 6.2.1 Condition: WBEM protocol supports client side control of returning class origin information.
IncludedProperties	PropertyName []	Optional	NULL, or unordered set of property names to be included, acting as a restricting filter on the properties included in the returned instances
ExcludeSubclassProperties	boolean	Optional	Indicates whether properties added by subclasses of the association class are to be excluded, acting as a restricting filter on the properties included in the returned instances
OperationTimeout	uint32	Mandatory	Operation timeout, as defined in 6.5.2.4
ContinueOnError	boolean	Conditional	Indicates whether the enumeration session should be continued in case of error, as defined in 6.5.2.5 Condition: WBEM protocol supports client side control of continuation on error for pulled instance enumeration operations.
MaxObjectCount	uint32	Mandatory	Maximum number of instances that may be returned by this operation, as defined in 6.5.2.6

1902

1903

Operation output parameters:

1904

Generic Name	Generic Type	Requirement	Description
InstanceList	InstanceSpecificationWithPath []	Mandatory	Sequence of the returned first set of instance representations and instance paths
EnumerationContext	EnumerationContext	Mandatory	Enumeration context value, as defined in 6.5.2.1

Generic Name	Generic Type	Requirement	Description
EndOfSequence	boolean	Mandatory	Indicates end of sequence for the enumeration session, as defined in 6.5.2.2

1905

1906 **Description:**

1907 The *OpenAssociators* operation establishes and opens an enumeration session for enumerating
 1908 instances that are associated with the specified source instance. That enumeration session allows
 1909 retrieving the instance representations and instance paths of these instances through successive
 1910 *PullInstancesWithPath* operations (see 6.5.11). Retrieval of a first set of those instances together
 1911 with their instance paths may be requested by setting *MaxObjectCount* to a value > 0.

1912 The set of instances to be enumerated throughout the entire enumeration session shall be
 1913 determined using the following algorithm:

1914 • Initially, the set of instances to be enumerated is the set of all instances associated to the
 1915 source instance referenced by *SourceInstancePath*. These associations may be instances
 1916 of different association classes. If the source instance does not exist, the operation shall
 1917 succeed with an empty result set (even when its creation class does not exist). However, if
 1918 the namespace of the source instance does not exist, the operation shall fail, indicating
 1919 WIPG0204.

1920 The result set should not contain any duplicate instances, as defined in 5.8.4. However,
 1921 different far ends may reference the same instance, and in such cases, the instance shall
 1922 be contained in the result set once for each such reference.

1923 • If the *AssociationClassName* operation input parameter is not NULL, it acts as a restricting
 1924 filter on the instances to be enumerated such that each instance that is associated with the
 1925 source instance using an association whose creation class or one of its superclasses does
 1926 not have the name specified in *AssociationClassName*, is removed from the set of
 1927 instances to be enumerated. There shall be no validity checking performed for the
 1928 *AssociationClassName* operation input parameter; if the specified class does not exist, the
 1929 operation shall succeed with an empty result (because the filter did not match).

1930 • If the *AssociatedClassName* operation input parameter is not NULL, it acts as a restricting
 1931 filter on the instances to be enumerated such that each instance whose creation class or
 1932 one of its superclasses does not have the name specified in *AssociatedClassName*, is
 1933 removed from the set of instances to be enumerated. There shall be no validity checking
 1934 performed for the *AssociatedClassName* operation input parameter; if the specified class
 1935 does not exist, the operation shall succeed with an empty result (because the filter did not
 1936 match).

1937 NOTE Specifying a non-NULL value for *AssociatedClassName* ensures that the returned instances
 1938 have the class specified in *AssociatedClassName* as a common superclass.

1939 • If the *SourceRoleName* operation input parameter is not NULL, it acts as a restricting filter
 1940 on the instances to be enumerated such that each instance that is associated with the
 1941 source instance using an association class that has a role name on the source end that is
 1942 not the role name specified in *SourceRoleName*, is removed from the set of instances to
 1943 be enumerated. There shall be no validity checking performed for the *SourceRoleName*
 1944 operation input parameter; if the specified role does not exist, the operation shall succeed
 1945 with an empty result (because the filter did not match).

1946 • If the *AssociatedRoleName* operation input parameter is not NULL, it acts as a restricting
 1947 filter on the instances to be enumerated such that each instance that is associated with the
 1948 source instance using an association class that has a role name on the end referencing

- 1949 that instance that is not the role name specified in *AssociatedRoleName*, is removed from
 1950 the set of instances to be enumerated. There shall be no validity checking performed for
 1951 the *AssociatedRoleName* operation input parameter; if the specified role does not exist, the
 1952 operation shall succeed with an empty result (because the filter did not match).
- 1953 • If the WBEM protocol supports filter queries for pulled instance enumeration operations
 1954 (that is, the *FilterQueryString* and *FilterQueryLanguage* operation parameters) and
 1955 *FilterQueryLanguage* is not NULL, *FilterQueryString* acts as a restricting filter on the
 1956 instances to be enumerated such that any instances not selected by the filter query for its
 1957 result set are removed from the set of instances. The filter query shall query only the class
 1958 specified in *AssociatedClassName* (e.g., in the CQL FROM-clause). See also 6.5.2.3.
- 1959 The set of instances to be enumerated throughout the entire enumeration session should not contain
 1960 any duplicate instances, as defined in 5.8.4. Because the set of returned instances contains only
 1961 instances that exist in the same namespace, a determination of duplicate instances can be done on
 1962 the basis of their model paths only.
- 1963 The set of instances to be returned (as instance representations and instance paths) is the first set of
 1964 instances from the set of instances to be enumerated throughout the entire enumeration session,
 1965 such that no more than *MaxObjectCount* instances are returned. Returning no instances does not
 1966 imply that the enumeration session has been exhausted. Only the *EndOfSequence* operation output
 1967 parameter indicates whether the enumeration session has been exhausted.
- 1968 The set of properties to be included in any returned instances shall be determined using the following
 1969 algorithm:
- 1970 • Initially, the set of properties to be included is the set of properties exposed by the creation
 1971 class of the instance. This includes all the duplicates of any duplicate non-overridden
 1972 properties.
 - 1973 • If the *IncludedProperties* operation input parameter is supported by the WBEM protocol
 1974 and if its value is not NULL, it acts as a restricting filter on the properties to be included in
 1975 the returned instances such that any properties exposed by the creation class of the
 1976 instance that are not named in that operation parameter are removed from the set of
 1977 properties to be included. Any duplicate or invalid property names in the
 1978 *IncludedProperties* operation input parameter shall be ignored. A non-NULL empty
 1979 *IncludedProperties* list removes all properties from the set of properties to be included.
 - 1980 • If the *ExcludeSubclassProperties* operation input parameter is supported by the WBEM
 1981 protocol and if its value is TRUE, it acts as a restricting filter on the properties to be
 1982 included in the returned instances such that any properties not exposed by the class
 1983 specified in *AssociatedClassName* are removed from the set of properties to be included.
 - 1984 • Conformant WBEM protocols may specify rules that cause properties with a value of NULL
 1985 to be removed from the set of properties to be included.
- 1986 **Preconditions:**
- 1987 • The namespace of the source instance referenced by *SourceInstancePath* shall exist. If it does
 1988 not exist, the operation shall fail, indicating WIPG0204.
 - 1989 • If a filter query is specified,
 - 1990 – the query language specified in the *FilterQueryLanguage* operation parameter shall be
 1991 valid. If this is not satisfied, the operation shall fail, indicating WIPG0221.
 - 1992 – the query specified in the *FilterQueryString* operation parameter shall be a valid query in
 1993 the query language specified in the *FilterQueryLanguage* operation parameter. If this is not
 1994 satisfied, the operation shall fail, indicating WIPG0222 or WIPG0223.

- 1995 – the *AssociatedClassName* operation input parameter shall be non-NULL. If this is not
1996 satisfied, the operation shall fail, indicating WIPG0208.
- 1997 • The *IncludedProperties* operation parameter, if supported by the WBEM protocol, shall only be
1998 specified with a non-NULL value if the *AssociatedClassName* operation input parameter is also
1999 non-NULL. If this is not satisfied, the operation shall fail, indicating WIPG0208.
- 2000 • The *ExcludeSubclassProperties* operation parameter, if supported by the WBEM protocol, shall
2001 only be specified with a TRUE value if the *AssociatedClassName* operation input parameter is
2002 non-NULL. If this is not satisfied, the operation shall fail, indicating WIPG0208.
- 2003 • The namespace of any returned instance paths shall exist. If it does not exist, the operation may
2004 fail, indicating WIPG0204. Note that cross-namespace association traversals may return
2005 instance paths in a server or namespace that is different from the server or namespace of the
2006 source instance.
- 2007 • The creation class of any returned instance paths shall exist in their namespace. If it does not
2008 exist, the operation may fail, indicating WIPG0214.

2009 **Postconditions:**

- 2010 • The enumeration session shall have been established and opened.
- 2011 • A first set of instances with their instance paths shall have been returned as described in the
2012 Description paragraph for this operation.
- 2013 • Requirements on ACID properties:
- 2014 – Atomicity: Required (related to the creation of an enumeration context that is maintained by
2015 the WBEM server)
- 2016 – Update Consistency: N/A
- 2017 – Isolation: Required at the level of single instances, as defined in 5.8.
- 2018 – Durability: Required (related to creation of an enumeration context that is maintained by
2019 the WBEM server)

2020 **Error Messages:**

2021

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM server is shutting down	Optional	Infrastructure	
WIPG0240	WBEM server limits are exceeded	Optional	Infrastructure, class implem.	
WIPG0204	Namespace not found	Mandatory	Infrastructure	For input namespace
WIPG0203	Operation not supported by WBEM server infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0242	Invalid timeout	Mandatory	Infrastructure, class implem.	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure, class implem.	
WIPG0235	Continuation on error not supported	Mandatory	Infrastructure, class implem.	
WIPG0237	Filter queries not supported by WBEM server infrastructure	Optional	Infrastructure	
WIPG0244	Filter queries not supported by class implementation	Optional	Class implem.	
WIPG0221	Unknown query language	Mandatory	Infrastructure, class implem.	
WIPG0222	Query language feature not supported	Mandatory	Infrastructure, class implem.	
WIPG0223	Invalid query	Mandatory	Infrastructure, class implem.	
WIPG0228	Operation not supported by class implementation	Mandatory	Class implem.	
WIPG0204	Namespace not found	Optional	Infrastructure	For namespace of returned instance paths
WIPG0214	Class not found	Optional	Infrastructure	
WIPG0243	Timeout	Optional	Infrastructure, class implem.	
WIPG0227	Other failure	Optional	Infrastructure, class implem.	

2022

2023 NOTE Version 1.1 of this specification removed WIPG0213 (Instance not found) from this table and added
 2024 WIPG0214 (Class not found).

2025 **6.5.6 OpenAssociatorPaths (deprecated)**

2026 **Purpose:**

2027 Establish and open an enumeration session for enumerating the instances that are associated with a
 2028 given source instance and optionally return a first set of their instance paths.

2029 **Operation input parameters:**

2030

Generic Name	Generic Type	Requirement	Description
SourceInstancePath	InstancePath	Mandatory	Instance path of the source instance (Context Parameter)
AssociationClassName	ClassName	Mandatory	NULL, or name of the association class, acting as a restricting filter on the returned instance paths

Generic Name	Generic Type	Requirement	Description
AssociatedClassName	ClassName	Mandatory	NULL, or name of the associated class on any far end of the association, acting as a restricting filter on the returned instance paths
SourceRoleName	PropertyName	Mandatory	NULL, or name of the role on the source end of the association, acting as a restricting filter on the returned instance paths
AssociatedRoleName	PropertyName	Mandatory	NULL, or name of the role on any far end of the association, acting as a restricting filter on the returned instance paths
FilterQueryString	QueryString	Conditional	NULL, or query string of a filter query that is acting as an additional restricting filter on the set of returned instance paths, as defined in 6.5.2.3 Condition: WBEM protocol supports filter queries for pulled instance enumeration operations.
FilterQueryLanguage	QueryLanguage	Conditional	NULL, or query language of the filter query specified in <i>FilterQueryString</i> , as defined in 6.5.2.3 Condition: WBEM protocol supports filter queries for pulled instance enumeration operations.
OperationTimeout	uint32	Mandatory	Operation timeout, as defined in 6.5.2.4
ContinueOnError	boolean	Conditional	Indicates whether the enumeration session should be continued in case of error, as defined in 6.5.2.5 Condition: WBEM protocol supports client side control of continuation on error for pulled instance enumeration operations.
MaxObjectCount	uint32	Mandatory	Maximum number of instances that may be returned by this operation, as defined in 6.5.2.6

2031

2032 **Operation output parameters:**

2033

Generic Name	Generic Type	Requirement	Description
InstancePathList	InstancePath []	Mandatory	Sequence of the returned first set of instance paths
EnumerationContext	EnumerationContext	Mandatory	Enumeration context value, as defined in 6.5.2.1
EndOfSequence	boolean	Mandatory	Indicates end of sequence for the enumeration session, as defined in 6.5.2.2

2034

2035 **Description:**

2036 The *OpenAssociatorPaths* operation establishes and opens an enumeration session for enumerating
 2037 the instance paths of instances that are associated with the specified source instance. That

- 2038 enumeration session allows retrieving the instance paths of these instances through successive
 2039 *PullInstancePaths* operations (see 6.5.12). Retrieval of a first set of those instance paths may be
 2040 requested by setting *MaxObjectCount* to a value > 0.
- 2041 The *OpenAssociatorPaths* operation has been deprecated in version 1.1.0 of this document. Use
 2042 *OpenAssociators* instead (see 6.5.5).
- 2043 The set of instances to be enumerated throughout the entire enumeration session shall be
 2044 determined using the following algorithm:
- 2045 • Initially, the set of instances to be enumerated is the set of all instances associated to the
 2046 source instance referenced by *SourceInstancePath*. These associations may be instances
 2047 of different association classes. If the source instance does not exist, the operation shall
 2048 succeed with an empty result set (even when its creation class does not exist). However, if
 2049 the namespace of the source instance does not exist, the operation shall fail, indicating
 2050 WIPG0204.
- 2051 The result set should not contain any duplicate instance paths, as defined in 5.8.4.
 2052 However, different far ends may reference the same instance, and in such cases, the
 2053 instance path shall be contained in the result set once for each such reference.
- 2054 • If the *AssociationClassName* operation input parameter is not NULL, it acts as a restricting
 2055 filter on the instances to be enumerated such that each instance that is associated with the
 2056 source instance using an association whose creation class or one of its superclasses does
 2057 not have the name specified in *AssociationClassName*, is removed from the set of
 2058 instances to be enumerated. There shall be no validity checking performed for the
 2059 *AssociationClassName* operation input parameter; if the specified class does not exist, the
 2060 operation shall succeed with an empty result (because the filter did not match).
 - 2061 • If the *AssociatedClassName* operation input parameter is not NULL, it acts as a restricting
 2062 filter on the instances to be enumerated such that each instance whose creation class or
 2063 one of its superclasses does not have the name specified in *AssociatedClassName*, is
 2064 removed from the set of instances to be enumerated. There shall be no validity checking
 2065 performed for the *AssociatedClassName* operation input parameter; if the specified class
 2066 does not exist, the operation shall succeed with an empty result (because the filter did not
 2067 match).
- 2068 NOTE Specifying a non-NULL value for *AssociatedClassName* ensures that the returned instances
 2069 have the class specified in *AssociatedClassName* as a common superclass.
- 2070 • If the *SourceRoleName* operation input parameter is not NULL, it acts as a restricting filter
 2071 on the instances to be enumerated such that each instance that is associated with the
 2072 source instance using an association class that has a role name on the source end that is
 2073 not the role name specified in *SourceRoleName*, is removed from the set of instances to
 2074 be enumerated. There shall be no validity checking performed for the *SourceRoleName*
 2075 operation input parameter; if the specified role does not exist, the operation shall succeed
 2076 with an empty result (because the filter did not match).

- 2077
- 2078
- 2079
- 2080
- 2081
- 2082
- 2083
- If the *AssociatedRoleName* operation input parameter is not NULL, it acts as a restricting filter on the instances to be enumerated such that each instance that is associated with the source instance using an association class that has a role name on the end referencing that instance that is not the role name specified in *AssociatedRoleName*, is removed from the set of instances to be enumerated. There shall be no validity checking performed for the *AssociatedRoleName* operation input parameter; if the specified role does not exist, the operation shall succeed with an empty result (because the filter did not match).
- 2084
- 2085
- 2086
- 2087
- 2088
- 2089
- If the WBEM protocol supports filter queries for pulled instance enumeration operations (that is, the *FilterQueryString* and *FilterQueryLanguage* operation parameters) and *FilterQueryLanguage* is not NULL, *FilterQueryString* acts as a restricting filter on the instances to be enumerated such that any instances not selected by the filter query for its result set are removed from the set of instances. The filter query shall query only the class specified in *AssociatedClassName* (e.g., in the CQL FROM-clause). See also 6.5.2.3.

2090

2091

2092

2093

The set of instances to be enumerated throughout the entire enumeration session should not contain any duplicate instance paths, as defined in 5.8.4. Because the set of returned instance paths references only instances in the same namespace, a determination of duplicate instance paths can be done on the basis of their model paths only.

2094

2095

2096

2097

2098

The set of instances to be returned (as instance paths) is the first set of instances from the set of instances to be enumerated throughout the entire enumeration session, such that no more than *MaxObjectCount* instances are returned. Returning no instances does not imply that the enumeration session has been exhausted. Only the *EndOfSequence* operation output parameter indicates whether the enumeration session has been exhausted.

2099

Preconditions:

- 2100
- 2101
- 2102
- 2103
- 2104
- 2105
- 2106
- 2107
- 2108
- 2109
- 2110
- 2111
- 2112
- 2113
- 2114
- 2115
- The namespace of the source instance referenced by *SourceInstancePath* shall exist. If it does not exist, the operation shall fail, indicating WIPG0204.
 - If a filter query is specified,
 - the query language specified in the *FilterQueryLanguage* operation parameter shall be valid. If this is not satisfied, the operation shall fail, indicating WIPG0221.
 - the query specified in the *FilterQueryString* operation parameter shall be a valid query in the query language specified in the *FilterQueryLanguage* operation parameter. If this is not satisfied, the operation shall fail, indicating WIPG0222 or WIPG0223.
 - the *AssociatedClassName* operation input parameter shall be non-NULL. If this is not satisfied, the operation shall fail, indicating WIPG0208.
 - The namespace of any returned instance paths shall exist. If it does not exist, the operation may fail, indicating WIPG0204. Note that cross-namespace association traversals may return instance paths in a server or namespace that is different from the server or namespace of the source instance.
 - The creation class of any returned instance paths shall exist in their namespace. If it does not exist, the operation may fail, indicating WIPG0214.

2116

Postconditions:

- 2117
- 2118
- 2119
- 2120
- The enumeration session shall have been established and opened.
 - A first set of instance paths shall have been returned as described in the Description paragraph for this operation.
 - Requirements on ACID properties:

- 2121 – Atomicity: Required (related to the creation of an enumeration context that is maintained by the WBEM server)
- 2122
- 2123 – Update Consistency: N/A
- 2124 – Isolation: Required at the level of single instances, as defined in 5.8.
- 2125 – Durability: Required (related to creation of an enumeration context that is maintained by the WBEM server)
- 2126

Error Messages:

2127
2128

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM server is shutting down	Optional	Infrastructure	
WIPG0240	WBEM server limits are exceeded	Optional	Infrastructure, class implem.	
WIPG0204	Namespace not found	Mandatory	Infrastructure	For input namespace
WIPG0203	Operation not supported by WBEM server infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0242	Invalid timeout	Mandatory	Infrastructure, class implem.	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure, class implem.	
WIPG0235	Continuation on error not supported	Mandatory	Infrastructure, class implem.	
WIPG0237	Filter queries not supported by WBEM server infrastructure	Optional	Infrastructure	
WIPG0244	Filter queries not supported by class implementation	Optional	Class implem.	
WIPG0221	Unknown query language	Mandatory	Infrastructure, class implem.	
WIPG0222	Query language feature not supported	Mandatory	Infrastructure, class implem.	
WIPG0223	Invalid query	Mandatory	Infrastructure, class implem.	
WIPG0228	Operation not supported by class implementation	Mandatory	Class implem.	
WIPG0204	Namespace not found	Optional	Infrastructure	For namespace of returned instance paths

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0214	Class not found	Optional	Infrastructure	
WIPG0243	Timeout	Optional	Infrastructure, class implem.	
WIPG0227	Other failure	Optional	Infrastructure, class implem.	

2129

2130 NOTE Version 1.1 of this specification removed WIPG0213 (Instance not found) from this table and added
2131 WIPG0214 (Class not found).

2132 6.5.7 OpenReferences

2133 Purpose:

2134 Establish and open an enumeration session for enumerating the association instances that reference
2135 a given source instance and optionally return a first set of their instance representations and instance
2136 paths.

2137 Operation input parameters:

2138

Generic Name	Generic Type	Requirement	Description
SourceInstancePath	InstancePath	Mandatory	Instance path of the source instance (Context Parameter)
AssociationClassName	ClassName	Mandatory	NULL, or name of the association class, acting as a restricting filter on the returned instances
SourceRoleName	PropertyName	Mandatory	NULL, or name of the role on the source end of the association, acting as a restricting filter on the returned instances
FilterQueryString	QueryString	Conditional	NULL, or query string of a filter query that is acting as an additional restricting filter on the set of returned instances, as defined in 6.5.2.3 Condition: WBEM protocol supports filter queries for pulled instance enumeration operations.
FilterQueryLanguage	QueryLanguage	Conditional	NULL, or query language of the filter query specified in <i>FilterQueryString</i> , as defined in 6.5.2.3 Condition: WBEM protocol supports filter queries for pulled instance enumeration operations.
IncludeClassOrigin	boolean	Conditional	Deprecated: Indicates whether class origin information for any returned properties is to be included, as defined in 6.2.1 Condition: WBEM protocol supports client side control of returning class origin information.

Generic Name	Generic Type	Requirement	Description
IncludedProperties	PropertyName []	Optional	NULL, or unordered set of property names to be included, acting as a restricting filter on the properties included in the returned instances
ExcludeSubclassProperties	boolean	Optional	Indicates whether properties added by subclasses of the association class are to be excluded, acting as a restricting filter on the properties included in the returned instances
OperationTimeout	uint32	Mandatory	Operation timeout, as defined in 6.5.2.4
ContinueOnError	boolean	Conditional	Indicates whether the enumeration session should be continued in case of error, as defined in 6.5.2.5 Condition: WBEM protocol supports client side control of continuation on error for pulled instance enumeration operations.
MaxObjectCount	uint32	Mandatory	Maximum number of instances that may be returned by this operation, as defined in 6.5.2.6

2139

2140 **Operation output parameters:**

2141

Generic Name	Generic Type	Requirement	Description
InstanceList	InstanceSpecificationWithPath []	Mandatory	Sequence of the returned first set of instance representations and instance paths
EnumerationContext	EnumerationContext	Mandatory	Enumeration context value, as defined in 6.5.2.1
EndOfSequence	boolean	Mandatory	Indicates end of sequence for the enumeration session, as defined in 6.5.2.2

2142

2143 **Description:**

2144 The *OpenReferences* operation establishes and opens an enumeration session for enumerating the
 2145 association instances that reference the specified source instance. That enumeration session allows
 2146 retrieving the instance representations and instance paths of these instances through successive
 2147 *PullInstancesWithPath* operations (see 6.5.11). Retrieval of a first set of those instances together
 2148 with their instance paths may be requested by setting *MaxObjectCount* to a value > 0.

2149 The set of instances to be enumerated throughout the entire enumeration session shall be
 2150 determined using the following algorithm:

- 2151 • Initially, the set of instances to be enumerated is the set of all instances referencing the
 2152 source instance referenced by *SourceInstancePath*. These associations may be instances
 2153 of different association classes. If the source instance does not exist, the operation shall
 2154 succeed with an empty result set (even when its creation class does not exist). However, if
 2155 the namespace of the source instance does not exist, the operation shall fail, indicating
 2156 WIPG0204.

- 2157 • If the *AssociationClassName* operation input parameter is not NULL, it acts as a restricting
2158 filter on the instances to be enumerated such that each association instance whose
2159 creation class or one of its superclasses does not have the name specified in
2160 *AssociationClassName*, is removed from the set of instances to be enumerated. There
2161 shall be no validity checking performed for the *AssociationClassName* operation input
2162 parameter; if the specified class does not exist, the operation shall succeed with an empty
2163 result (because the filter did not match).
- 2164 NOTE Specifying a non-NULL value for *AssociationClassName* ensures that the returned
2165 instances have the class specified in *AssociationClassName* as a common superclass.
- 2166 • If the *SourceRoleName* operation input parameter is not NULL, it acts as a restricting filter
2167 on the instances to be enumerated such that each association instance whose creation
2168 class does not have the role name specified in *SourceRoleName* on the end referencing
2169 the source instance, is removed from the set of instances to be enumerated. There shall be
2170 no validity checking performed for the *SourceRoleName* operation input parameter; if the
2171 specified role does not exist, the operation shall succeed with an empty result (because the
2172 filter did not match).
- 2173 • If the WBEM protocol supports filter queries for pulled instance enumeration operations
2174 (that is, the *FilterQueryString* and *FilterQueryLanguage* operation parameters) and
2175 *FilterQueryLanguage* is not NULL, *FilterQueryString* acts as a restricting filter on the
2176 instances to be enumerated such that any instances not selected by the filter query for its
2177 result set are removed from the set of instances. The filter query shall query only the class
2178 specified in *AssociationClassName* (e.g., in the CQL FROM-clause). See also 6.5.2.3.
- 2179 NOTE Version 1.1 of this specification removed the *AssociatedClassName* and *AssociatedRoleName* filters
2180 from this operation.
- 2181 The set of instances to be enumerated throughout the entire enumeration session should not contain
2182 any duplicate instances, as defined in 5.8.4. Because the set of returned instances contains only
2183 instances that exist in the same namespace, so any determination of duplicate instances (for
2184 example by a WBEM client) may be done on the basis of their model paths.
- 2185 The set of instances to be returned (as instance representations and instance paths) is the first set of
2186 instances from the set of instances to be enumerated throughout the entire enumeration session,
2187 such that no more than *MaxObjectCount* instances are returned. Returning no instances does not
2188 imply that the enumeration session has been exhausted. Only the *EndOfSequence* operation output
2189 parameter indicates whether the enumeration session has been exhausted.
- 2190 The set of properties to be included in any returned instances shall be determined using the following
2191 algorithm:
- 2192 • Initially, the set of properties to be included is the set of properties exposed by the creation
2193 class of the instance. This includes all the duplicates of any duplicate non-overridden
2194 properties.
- 2195 • If the *IncludedProperties* operation input parameter is supported by the WBEM protocol
2196 and if its value is not NULL, it acts as a restricting filter on the properties to be included in
2197 the returned instances such that any properties exposed by the creation class of the
2198 instance that are not named in that operation parameter are removed from the set of
2199 properties to be included. Any duplicate or invalid property names in the
2200 *IncludedProperties* operation input parameter shall be ignored. A non-NULL empty
2201 *IncludedProperties* list removes all properties from the set of properties to be included.

- 2202 • If the *ExcludeSubclassProperties* operation input parameter is supported by the WBEM
2203 protocol and if its value is TRUE, it acts as a restricting filter on the properties to be
2204 included in the returned instances such that any properties not exposed by the class
2205 specified in *AssociationClassName* are removed from the set of properties to be included.
- 2206 • Conformant WBEM protocols may specify rules that cause properties with a value of NULL
2207 to be removed from the set of properties to be included.

2208 **Preconditions:**

- 2209 • The namespace of the source instance referenced by *SourceInstancePath* shall exist. If it does
2210 not exist, the operation shall fail, indicating WIPG0204.
- 2211 • If a filter query is specified,
 - 2212 – the query language specified in the *FilterQueryLanguage* operation parameter shall be
2213 valid. If this is not satisfied, the operation shall fail, indicating WIPG0221.
 - 2214 – the query specified in the *FilterQueryString* operation parameter shall be a valid query in
2215 the query language specified in the *FilterQueryLanguage* operation parameter. If this is not
2216 satisfied, the operation shall fail, indicating WIPG0222 or WIPG0223.
 - 2217 – the *AssociationClassName* operation input parameter shall be non-NULL. If this is not
2218 satisfied, the operation shall fail, indicating WIPG0208.
- 2219 • The *IncludedProperties* operation parameter, if supported by the WBEM protocol, shall only be
2220 specified with a non-NULL value if the *AssociationClassName* operation input parameter is also
2221 non-NULL. If this is not satisfied, the operation shall fail, indicating WIPG0208.
- 2222 • The *ExcludeSubclassProperties* operation parameter, if supported by the WBEM protocol, shall
2223 only be specified with a TRUE value if the *AssociationClassName* operation input parameter is
2224 non-NULL. If this is not satisfied, the operation shall fail, indicating WIPG0208.
- 2225 • The namespace of any returned instance paths shall exist. If it does not exist, the operation may
2226 fail, indicating WIPG0204. Note that cross-namespace association traversals may return
2227 instance paths in a server or namespace that is different from the server or namespace of the
2228 source instance.
- 2229 • The creation class of any returned instance paths shall exist in their namespace. If it does not
2230 exist, the operation may fail, indicating WIPG0214.

2231 **Postconditions:**

- 2232 • The enumeration session shall have been established and opened.
- 2233 • A first set of instances with their instance paths shall have been returned as described in the
2234 Description paragraph for this operation.
- 2235 • Requirements on ACID properties:
 - 2236 – Atomicity: Required (related to the creation of an enumeration context that is maintained by
2237 the WBEM server)
 - 2238 – Update Consistency: N/A
 - 2239 – Isolation: Required at the level of single instances, as defined in 5.8.
 - 2240 – Durability: Required (related to creation of an enumeration context that is maintained by
2241 the WBEM server)

2242
2243**Error Messages:**

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM server is shutting down	Optional	Infrastructure	
WIPG0240	WBEM server limits are exceeded	Optional	Infrastructure, class implem.	
WIPG0204	Namespace not found	Mandatory	Infrastructure	For input namespace
WIPG0203	Operation not supported by WBEM server infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0242	Invalid timeout	Mandatory	Infrastructure, class implem.	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure, class implem.	
WIPG0235	Continuation on error not supported	Mandatory	Infrastructure, class implem.	
WIPG0237	Filter queries not supported by WBEM server infrastructure	Optional	Infrastructure	
WIPG0244	Filter queries not supported by class implementation	Optional	Class implem.	
WIPG0221	Unknown query language	Mandatory	Infrastructure, class implem.	
WIPG0222	Query language feature not supported	Mandatory	Infrastructure, class implem.	
WIPG0223	Invalid query	Mandatory	Infrastructure, class implem.	
WIPG0228	Operation not supported by class implementation	Mandatory	Class implem.	
WIPG0204	Namespace not found	Optional	Infrastructure	For namespace of returned instance paths
WIPG0214	Class not found	Optional	Infrastructure	
WIPG0243	Timeout	Optional	Infrastructure, class implem.	
WIPG0227	Other failure	Optional	Infrastructure, class implem.	

2244

2245
2246

NOTE Version 1.1 of this specification removed WIPG0213 (Instance not found) from this table and added WIPG0214 (Class not found).

2247 **6.5.8 OpenReferencePaths (deprecated)**

2248 **Purpose:**

2249 Establish and open an enumeration session for enumerating the association instances that reference
 2250 a given source instance and optionally return a first set of their instance paths.

2251 **Operation input parameters:**

2252

Generic Name	Generic Type	Requirement	Description
SourceInstancePath	InstancePath	Mandatory	Instance path of the source instance (Context Parameter)
AssociationClassName	ClassName	Mandatory	NULL, or name of the association class, acting as a restricting filter on the returned instance paths
SourceRoleName	PropertyName	Mandatory	NULL, or name of the role on the source end of the association, acting as a restricting filter on the returned instance paths
FilterQueryString	QueryString	Conditional	NULL, or query string of a filter query that is acting as an additional restricting filter on the set of returned instance paths, as defined in 6.5.2.3 Condition: WBEM protocol supports filter queries for pulled instance enumeration operations.
FilterQueryLanguage	QueryLanguage	Conditional	NULL, or query language of the filter query specified in <i>FilterQueryString</i> , as defined in 6.5.2.3 Condition: WBEM protocol supports filter queries for pulled instance enumeration operations.
OperationTimeout	uint32	Mandatory	Operation timeout, as defined in 6.5.2.4
ContinueOnError	boolean	Conditional	Indicates whether the enumeration session should be continued in case of error, as defined in 6.5.2.5 Condition: WBEM protocol supports client side control of continuation on error for pulled instance enumeration operations.
MaxObjectCount	uint32	Mandatory	Maximum number of instance paths that may be returned by this operation, as defined in 6.5.2.6

2253

2254 **Operation output parameters:**

2255

Generic Name	Generic Type	Requirement	Description
InstancePathList	InstancePath []	Mandatory	Sequence of the returned first set of instance paths
EnumerationContext	EnumerationContext	Mandatory	Enumeration context value, as defined in 6.5.2.1

Generic Name	Generic Type	Requirement	Description
EndOfSequence	boolean	Mandatory	Indicates end of sequence for the enumeration session, as defined in 6.5.2.2

2256

2257 **Description:**

2258 The *OpenReferencePaths* operation establishes and opens an enumeration session for enumerating
 2259 the instance paths of association instances that reference the specified source instance. That
 2260 enumeration session allows retrieving the instance paths of these instances through successive
 2261 *PullInstancePaths* operations (see 6.5.12). Retrieval of a first set of those instance paths may be
 2262 requested by setting *MaxObjectCount* to a value > 0.

2263 The *OpenReferencePaths* operation has been deprecated in version 1.1.0 of this document. Use
 2264 *OpenReferences* instead (see 6.5.7).

2265 The set of instances to be enumerated throughout the entire enumeration session shall be
 2266 determined using the following algorithm:

2267

- Initially, the set of instances to be enumerated is the set of all instances referencing the
 2268 source instance referenced by *SourceInstancePath*. These associations may be instances
 2269 of different association classes. If the source instance does not exist, the operation shall
 2270 succeed with an empty result set (even when its creation class does not exist). However, if
 2271 the namespace of the source instance does not exist, the operation shall fail, indicating
 2272 WIPG0204.

2273

- If the *AssociationClassName* operation input parameter is not NULL, it acts as a restricting
 2274 filter on the instances to be enumerated such that each association instance whose
 2275 creation class or one of its superclasses does not have the name specified in
 2276 *AssociationClassName*, is removed from the set of instances to be enumerated. There
 2277 shall be no validity checking performed for the *AssociationClassName* operation input
 2278 parameter; if the specified class does not exist, the operation shall succeed with an empty
 2279 result (because the filter did not match).

2280 NOTE Specifying a non-NULL value for *AssociationClassName* ensures that the returned
 2281 instances have the class specified in *AssociationClassName* as a common superclass.

2282

- If the *SourceRoleName* operation input parameter is not NULL, it acts as a restricting filter
 2283 on the instances to be enumerated such that each association instance whose creation
 2284 class does not have the role name specified in *SourceRoleName* on the end referencing
 2285 the source instance, is removed from the set of instances to be enumerated. There shall be
 2286 no validity checking performed for the *SourceRoleName* operation input parameter; if the
 2287 specified role does not exist, the operation shall succeed with an empty result (because the
 2288 filter did not match).

2289

- If the WBEM protocol supports filter queries for pulled instance enumeration operations
 2290 (that is, the *FilterQueryString* and *FilterQueryLanguage* operation parameters) and
 2291 *FilterQueryLanguage* is not NULL, *FilterQueryString* acts as a restricting filter on the
 2292 instances to be enumerated such that any instances not selected by the filter query for its
 2293 result set are removed from the set of instances. The filter query shall query only the class
 2294 specified in *AssociationClassName* (e.g., in the CQL FROM-clause). See also 6.5.2.3.

2295 NOTE Version 1.1 of this specification removed the *AssociatedClassName* and *AssociatedRoleName* filters
 2296 from this operation.

2297 The set of instance paths to be enumerated throughout the entire enumeration session should not
 2298 contain any duplicate instance paths, as defined in 5.8.4. Because the set of returned instance paths

2299 references only instances that exist in the same namespace, a determination of duplicate instance
 2300 paths can be done on the basis of their model paths only.

2301 The set of instances to be returned (as instance paths) is the first set of instances from the set of
 2302 instances to be enumerated throughout the entire enumeration session, such that no more than
 2303 *MaxObjectCount* instances are returned. Returning no instances does not imply that the enumeration
 2304 session has been exhausted. Only the *EndOfSequence* operation output parameter indicates
 2305 whether the enumeration session has been exhausted.

2306 **Preconditions:**

- 2307 • The namespace of the source instance referenced by *SourceInstancePath* shall exist. If it does
 2308 not exist, the operation shall fail, indicating WIPG0204.
- 2309 • If a filter query is specified,
 - 2310 – the query language specified in the *FilterQueryLanguage* operation parameter shall be
 2311 valid. If this is not satisfied, the operation shall fail, indicating WIPG0221.
 - 2312 – the query specified in the *FilterQueryString* operation parameter shall be a valid query in
 2313 the query language specified in the *FilterQueryLanguage* operation parameter. If this is not
 2314 satisfied, the operation shall fail, indicating WIPG0222 or WIPG0223.
 - 2315 – the *AssociationClassName* operation input parameter shall be non-NULL. If this is not
 2316 satisfied, the operation shall fail, indicating WIPG0208.
- 2317 • The namespace of any returned instance paths shall exist. If it does not exist, the operation may
 2318 fail, indicating WIPG0204. Note that cross-namespace association traversals may return
 2319 instance paths in a server or namespace that is different from the server or namespace of the
 2320 source instance.
- 2321 • The creation class of any returned instance paths shall exist in their namespace. If it does not
 2322 exist, the operation may fail, indicating WIPG0214.

2323 **Postconditions:**

- 2324 • The enumeration session shall have been established and opened.
- 2325 • A first set of instance paths shall have been returned as described in the Description paragraph
 2326 for this operation.
- 2327 • Requirements on ACID properties:
 - 2328 – Atomicity: Required (related to the creation of an enumeration context that is maintained by
 2329 the WBEM server)
 - 2330 – Update Consistency: N/A
 - 2331 – Isolation: Required at the level of single instances, as defined in 5.8.
 - 2332 – Durability: Required (related to creation of an enumeration context that is maintained by
 2333 the WBEM server)

2334 **Error Messages:**

2335

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM server is shutting down	Optional	Infrastructure	

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0240	WBEM server limits are exceeded	Optional	Infrastructure, class implem.	
WIPG0204	Namespace not found	Mandatory	Infrastructure	For input namespace
WIPG0203	Operation not supported by WBEM server infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0242	Invalid timeout	Mandatory	Infrastructure, class implem.	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure, class implem.	
WIPG0235	Continuation on error not supported	Mandatory	Infrastructure, class implem.	
WIPG0237	Filter queries not supported by WBEM server infrastructure	Optional	Infrastructure	
WIPG0244	Filter queries not supported by class implementation	Optional	Class implem.	
WIPG0221	Unknown query language	Mandatory	Infrastructure, class implem.	
WIPG0222	Query language feature not supported	Mandatory	Infrastructure, class implem.	
WIPG0223	Invalid query	Mandatory	Infrastructure, class implem.	
WIPG0228	Operation not supported by class implementation	Mandatory	Class implem.	
WIPG0204	Namespace not found	Optional	Infrastructure	For namespace of returned instance paths
WIPG0214	Class not found	Optional	Infrastructure	
WIPG0243	Timeout	Optional	Infrastructure, class implem.	
WIPG0227	Other failure	Optional	Infrastructure, class implem.	

2336

2337 NOTE Version 1.1 of this specification removed WIPG0213 (Instance not found) from this table and added
 2338 WIPG0214 (Class not found).

2339 **6.5.9 OpenQueryInstances**

2340 **Purpose:**

2341 Establish and open an enumeration session for enumerating the instances of a query result in a
 2342 given namespace and optionally return a first set of their instance representations.

2343 **Operation input parameters:**

2344

Generic Name	Generic Type	Requirement	Description
NamespacePath	NamespacePath	Mandatory	Namespace path of the namespace in which the query is executed (Context Parameter)
QueryString	QueryString	Mandatory	Query string of a query that defines the set of instances to be returned
QueryLanguage	QueryLanguage	Mandatory	Query language of the query specified in <i>QueryString</i>
ReturnQueryResultClass	boolean	Mandatory	Indicates whether a class definition of the query result should be returned in <i>QueryResultClass</i>
OperationTimeout	uint32	Mandatory	Operation timeout, as defined in 6.5.2.4
ContinueOnError	boolean	Conditional	Indicates whether the enumeration session should be continued in case of error, as defined in 6.5.2.5 Condition: WBEM protocol supports client side control of continuation on error for pulled instance enumeration operations.
MaxObjectCount	uint32	Mandatory	Maximum number of instances that may be returned by this operation, as defined in 6.5.2.6

2345

2346 **Operation output parameters:**

2347

Generic Name	Generic Type	Requirement	Description
InstanceList	InstanceSpecification []	Mandatory	Sequence of the returned first set of instance representations
QueryResultClass	ClassSpecification	Mandatory	Representation of a class definition for the query result
EnumerationContext	EnumerationContext	Mandatory	Enumeration context value, as defined in 6.5.2.1
EndOfSequence	boolean	Mandatory	Indicates end of sequence for the enumeration session, as defined in 6.5.2.2

2348

2349 **Description:**

2350 The *OpenQueryInstances* operation establishes and opens an enumeration session for enumerating
 2351 the instances representing the result of the query specified in *QueryString* in the namespace
 2352 referenced by *NamespacePath*. That enumeration session allows retrieving representations of these
 2353 result instances through successive *PullInstances* operations (see 6.5.13). Retrieval of a first set of
 2354 those instances may be requested by setting *MaxObjectCount* to a value > 0.

2355 The set of instances to be returned (as instance representations) is the first set of instances from the
 2356 set of instances to be enumerated throughout the entire enumeration session, such that no more
 2357 than *MaxObjectCount* instances are returned. Returning no instances in the *InstanceList* operation
 2358 parameter does not imply that the enumeration session has been exhausted. Only the
 2359 *EndOfSequence* operation output parameter indicates whether the enumeration session has been
 2360 exhausted.

2361 The returned instance representations have no corresponding addressable instances that exist.

2362 If *QueryLanguage* is not NULL, it shall specify a valid query language and *QueryString* shall be a
 2363 valid query in that query language. Neither the query language nor the format of the filter query is
 2364 defined by this specification. Conformant WBEM protocols shall specify a mechanism for determining
 2365 the set of query languages that are valid for *QueryLanguage*. The simplest way to do this is to list the
 2366 set of valid query languages.

2367 The value of the *ReturnQueryResultClass* operation input parameter controls whether or not a class
 2368 definition is returned in the *QueryResultClass* operation output parameter. If FALSE, then
 2369 *QueryResultClass* shall be NULL. If TRUE, then the value of *QueryResultClass* shall be a class
 2370 definition that defines the properties of each instance of the query result. The name of this class shall
 2371 be CIM_QueryResult. This class is only a representation of a class that has no corresponding
 2372 addressable class residing in the WBEM server.

2373 **Preconditions:**

- 2374 • The namespace referenced by *NamespacePath* shall exist. If it does not exist, the operation
 2375 shall fail, indicating WIPG0204.
- 2376 • The query language specified in the *QueryLanguage* operation parameter shall be a valid query
 2377 language. If this is not satisfied, the operation shall fail, indicating WIPG0221.
- 2378 • The query specified in the *QueryString* operation parameter shall be a valid query in the query
 2379 language specified in the *QueryLanguage* operation parameter. If this is not satisfied, the
 2380 operation shall fail, indicating WIPG0222 or WIPG0223.

2381 **Postconditions:**

- 2382 • The enumeration session shall have been established and opened.
- 2383 • A first set of instances shall have been returned as described in the Description paragraph for
 2384 this operation.
- 2385 • Requirements on ACID properties:
 - 2386 – Atomicity: Required (related to the creation of an enumeration context that is maintained by
 2387 the WBEM server)
 - 2388 – Update Consistency: N/A
 - 2389 – Isolation: Required at the level of single instances, as defined in 5.8.
 - 2390 – Durability: Required (related to creation of an enumeration context that is maintained by
 2391 the WBEM server)

2392 **Error Messages:**

2393

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM server is shutting down	Optional	Infrastructure	
WIPG0240	WBEM server limits are exceeded	Optional	Infrastructure, class implem.	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM server infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0242	Invalid timeout	Mandatory	Infrastructure, class implem.	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure, class implem.	
WIPG0235	Continuation on error not supported	Mandatory	Infrastructure, class implem.	
WIPG0221	Unknown query language	Mandatory	Infrastructure, class implem.	
WIPG0222	Query language feature not supported	Mandatory	Infrastructure, class implem.	
WIPG0223	Invalid query	Mandatory	Infrastructure, class implem.	
WIPG0228	Operation not supported by class implementation	Mandatory	Class implem.	
WIPG0243	Timeout	Optional	Infrastructure, class implem.	
WIPG0227	Other failure	Optional	Infrastructure, class implem.	

2394

2395 **6.5.10 Common operation parameters for the pull operations**

2396 This subclause defines commonly used operation parameters for the Pull operations. The description of
 2397 the individual Pull operations references these operation parameters as appropriate. However, not every
 2398 Pull operation uses every one of these common operation parameters.

2399 **6.5.10.1 NamespacePath**

2400 The *NamespacePath* operation input parameter references the namespace identified by the context
 2401 parameter of the Open operation that established and opened the enumeration session.

2402 **6.5.10.2 EnumerationContext**

2403 The *EnumerationContext* operation input/output parameter is the enumeration context value representing
2404 the enumeration session to be used.

2405 Support for the *EnumerationContext* operation parameter in a conformant WBEM protocol is mandatory.

2406 When invoking the Pull operation, the enumeration session represented by *EnumerationContext* shall be
2407 open. The enumeration session shall have been established using one of the Open operations whose
2408 type of enumerated element matches the Pull operation. For the first Pull operation on an enumeration
2409 session, the value of *EnumerationContext* shall be the enumeration context value returned by a
2410 successful Open operation that established and opened that enumeration session. For any subsequent
2411 Pull operations on that enumeration session, the value of *EnumerationContext* shall be the value of
2412 *EnumerationContext* as returned by the previous Pull operation on the same enumeration session.

2413 After completing the Pull operation, the enumeration session represented by *EnumerationContext* shall
2414 be open or closed.

2415 **6.5.10.3 EndOfSequence**

2416 The *EndOfSequence* operation output parameter when used in Pull operations behaves as defined in
2417 6.5.2.2

2418 **6.5.10.4 MaxObjectCount**

2419 The *MaxObjectCount* operation input parameter when used in Pull operations behaves as defined in
2420 6.5.2.6.

2421 **6.5.11 PullInstancesWithPath**2422 **Purpose:**

2423 Retrieve the next set of instance representations and instance paths from an open enumeration
2424 session.

2425 **Operation input parameters:**

2426

Generic Name	Generic Type	Requirement	Description
NamespacePath	NamespacePath	Mandatory	Namespace path of the namespace for the enumeration, as defined in 6.5.10.1 (Context Parameter)
EnumerationContext	EnumerationContext	Mandatory	Enumeration context value, as defined in 6.5.10.2
MaxObjectCount	uint32	Mandatory	Maximum number of instances that may be returned by this operation, as defined in 6.5.10.4

2427

2428 **Operation output parameters:**

2429

Generic Name	Generic Type	Requirement	Description
InstanceList	InstanceSpecificationWithPath []	Mandatory	Next set of returned instance representations and instance paths

Generic Name	Generic Type	Requirement	Description
EnumerationContext	EnumerationContext	Mandatory	Enumeration context value, as defined in 6.5.10.2
EndOfSequence	boolean	Mandatory	Indicates end of sequence for the enumeration session, as defined in 6.5.10.3

2430

2431 **Description:**

2432 The *PullInstancesWithPath* operation retrieves the next set of instance representations and instance
2433 paths from an open enumeration session.

2434 The enumeration session shall have been established using one of the following operations:

- 2435 • OpenEnumerateInstances
- 2436 • OpenAssociators
- 2437 • OpenReferences

2438 The set of instances to be returned (as instance representations and instance paths) is the next set
2439 of instances from the set of instances to be enumerated throughout the entire enumeration session,
2440 such that no more than *MaxObjectCount* instances are returned. Returning no instances does not
2441 imply that the enumeration session has been exhausted. Only the *EndOfSequence* operation output
2442 parameter indicates whether the enumeration session has been exhausted.

2443 The set of properties to be included in any retrieved instances shall be the as determined using the
2444 Open operation that established the enumeration session.

2445 **Preconditions:**

- 2446 • The namespace referenced by *NamespacePath* shall exist. If it does not exist, the operation
2447 shall fail, indicating WIPG0204.
- 2448 • The enumeration session identified by *EnumerationContext* shall be open. If this is not satisfied,
2449 the operation shall fail, indicating WIPG0241.
- 2450 • The value of *EnumerationContext* shall be the enumeration context value returned by the
2451 previous Open or Pull operation on the same enumeration session. If this is not satisfied, the
2452 operation shall fail, indicating WIPG0241.
- 2453 • The namespace of any returned instance paths shall exist. If it does not exist, the operation may
2454 fail, indicating WIPG0204. Note that cross-namespace association traversals may return
2455 instance paths in a server or namespace that is different from the server or namespace of the
2456 source instance.
- 2457 • The creation class of any returned instance paths shall exist in their namespace. If it does not
2458 exist, the operation may fail, indicating WIPG0214.

2459 **Postconditions:**

- 2460 • The set of instances with their instance paths shall have been returned as described in the
2461 Description paragraph for this operation.
- 2462 • Requirements on ACID properties:
 - 2463 – Atomicity: Required (related to updates to an enumeration context that is maintained by the
2464 WBEM server)

- 2465 – Update Consistency: N/A
- 2466 – Isolation: Required at the level of single instances, as defined in 5.8.
- 2467 – Durability: Required (related to updates to an enumeration context that is maintained by
- 2468 the WBEM server)

2469 **Error Messages:**

2470

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM server is shutting down	Optional	Infrastructure	
WIPG0240	WBEM server limits are exceeded	Optional	Infrastructure, class implem.	
WIPG0203	Operation not supported by WBEM server infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure, class implem.	
WIPG0228	Operation not supported by class implementation	Mandatory	Class implem.	
WIPG0241	Invalid enumeration context	Mandatory	Class implem.	
WIPG0238	Pull operation has been abandoned due to enumeration context closure	Mandatory	Class implem.	
WIPG0243	Timeout	Optional	Infrastructure, class implem.	
WIPG0227	Other failure	Optional	Infrastructure, class implem.	

2471

2472 **6.5.12 PullInstancePaths (deprecated)**2473 **Purpose:**

2474 Retrieve the next set of instance paths from an open enumeration session.

2475 **Operation input parameters:**

2476

Generic Name	Generic Type	Requirement	Description
NamespacePath	NamespacePath	Mandatory	Namespace path of the namespace for the enumeration, as defined in 6.5.10.1 (Context Parameter)

Generic Name	Generic Type	Requirement	Description
EnumerationContext	EnumerationContext	Mandatory	Enumeration context value, as defined in 6.5.10.2
MaxObjectCount	uint32	Mandatory	Maximum number of instance paths that may be returned by this operation, as defined in 6.5.10.4

2477

2478 **Operation output parameters:**

2479

Generic Name	Generic Type	Requirement	Description
InstancePathList	InstancePath []	Mandatory	Next set of returned instance paths
EnumerationContext	EnumerationContext	Mandatory	Enumeration context value, as defined in 6.5.10.2
EndOfSequence	boolean	Mandatory	Indicates end of sequence for the enumeration session, as defined in 6.5.10.3

2480

2481 **Description:**

2482 The *PullInstancePaths* operation retrieves the next set of instance paths from an open enumeration
2483 session.

2484 The *PullInstancePaths* operation has been deprecated in version 1.1.0 of this document, because all
2485 operations that return instance paths from an open enumeration session have been deprecated.

2486 The enumeration session shall have been established using one of the following operations:

- 2487 • OpenEnumerateInstancePaths (deprecated)
- 2488 • OpenAssociatorPaths (deprecated)
- 2489 • OpenReferencePaths (deprecated)

2490 The set of instances to be returned (as instance paths) is the next set of instances from the set of
2491 instances to be enumerated throughout the entire enumeration session, such that no more than
2492 *MaxObjectCount* instance paths are returned. Returning no instance paths does not imply that the
2493 enumeration session has been exhausted. Only the *EndOfSequence* operation output parameter
2494 indicates whether the enumeration session has been exhausted.

2495 **Preconditions:**

- 2496 • The namespace referenced by *NamespacePath* shall exist. If it does not exist, the operation
2497 shall fail, indicating WIPG0204.
- 2498 • The enumeration session identified by *EnumerationContext* shall be open. If this is not satisfied,
2499 the operation shall fail, indicating WIPG0241.
- 2500 • The value of *EnumerationContext* shall be the enumeration context value returned by the
2501 previous Open or Pull operation on the same enumeration session. If this is not satisfied, the
2502 operation shall fail, indicating WIPG0241.
- 2503 • The namespace of any returned instance paths shall exist. If it does not exist, the operation may
2504 fail, indicating WIPG0204. Note that cross-namespace association traversals may return

2505 instance paths in a server or namespace that is different from the server or namespace of the
2506 source instance.

- 2507 • The creation class of any returned instance paths shall exist in their namespace. If it does not
2508 exist, the operation may fail, indicating WIPG0214.

2509 Postconditions:

- 2510 • The set of instance paths shall have been returned as described in the Description paragraph
2511 for this operation.
- 2512 • Requirements on ACID properties:
 - 2513 – Atomicity: Required (related to updates to an enumeration context that is maintained by the
2514 WBEM server)
 - 2515 – Update Consistency: N/A
 - 2516 – Isolation: Required at the level of single instances, as defined in 5.8.
 - 2517 – Durability: Required (related to updates to an enumeration context that is maintained by
2518 the WBEM server)

2519 Error Messages:

2520

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM server is shutting down	Optional	Infrastructure	
WIPG0240	WBEM server limits are exceeded	Optional	Infrastructure, class implem.	
WIPG0203	Operation not supported by WBEM server infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure, class implem.	
WIPG0228	Operation not supported by class implementation	Mandatory	Class implem.	
WIPG0241	Invalid enumeration context	Mandatory	Class implem.	
WIPG0238	Pull operation has been abandoned due to enumeration context closure	Mandatory	Class implem.	
WIPG0243	Timeout	Optional	Infrastructure, class implem.	
WIPG0227	Other failure	Optional	Infrastructure, class implem.	

2521

2522 **6.5.13 PullInstances**

2523 **Purpose:**

2524 Retrieve the next set of instances from an open enumeration session.

2525 **Operation input parameters:**

2526

Generic Name	Generic Type	Requirement	Description
NamespacePath	NamespacePath	Mandatory	Namespace path of the namespace for the enumeration, as defined in 6.5.10.1 (Context Parameter)
EnumerationContext	EnumerationContext	Mandatory	Enumeration context value, as defined in 6.5.10.2
MaxObjectCount	uint32	Mandatory	Maximum number of instances that may be returned by this operation, as defined in 6.5.10.4

2527

2528 **Operation output parameters:**

2529

Generic Name	Generic Type	Requirement	Description
InstanceList	InstanceSpecification []	Mandatory	Next set of returned instance representations
EnumerationContext	EnumerationContext	Mandatory	Enumeration context value, as defined in 6.5.10.2
EndOfSequence	boolean	Mandatory	Indicates end of sequence for the enumeration session, as defined in 6.5.10.3

2530

2531 **Description:**

2532 The *PullInstances* operation retrieves the next set of instance representations without their instance
2533 paths from an open enumeration session.

2534 The enumeration session shall have been established using one of the following operations:

- 2535 • *OpenQueryInstances*

2536 The set of instances to be returned (as instance representations) is the next set of instances from the
2537 set of instances to be enumerated throughout the entire enumeration session, such that no more
2538 than *MaxObjectCount* instances are returned. Returning no instances does not imply that the
2539 enumeration session has been exhausted. Only the *EndOfSequence* operation output parameter
2540 indicates whether the enumeration session has been exhausted.

2541 The set of properties to be included in any retrieved instances shall be the as determined using the
2542 Open operation that established the enumeration session.

2543 **Preconditions:**

- 2544 • The namespace referenced by *NamespacePath* shall exist. If it does not exist, the operation
2545 shall fail, indicating WIPG0204.
- 2546 • The enumeration session identified by *EnumerationContext* shall be open. If this is not satisfied,
2547 the operation shall fail, indicating WIPG0241.
- 2548 • The value of *EnumerationContext* shall be the enumeration context value returned by the
2549 previous Open or Pull operation on the same enumeration session. If this is not satisfied, the
2550 operation shall fail, indicating WIPG0241.

2551 **Postconditions:**

- 2552 • The set of instances shall have been returned as described in the Description paragraph for this
2553 operation.
- 2554 • Requirements on ACID properties:
 - 2555 – Atomicity: Required (related to updates to an enumeration context that is maintained by the
2556 WBEM server)
 - 2557 – Update Consistency: N/A
 - 2558 – Isolation: Required at the level of single instances, as defined in 5.8.
 - 2559 – Durability: Required (related to updates to an enumeration context that is maintained by
2560 the WBEM server)

2561 **Error Messages:**

2562

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM server is shutting down	Optional	Infrastructure	
WIPG0240	WBEM server limits are exceeded	Optional	Infrastructure, class implem.	
WIPG0203	Operation not supported by WBEM server infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure, class implem.	
WIPG0228	Operation not supported by class implementation	Mandatory	Class implem.	
WIPG0241	Invalid enumeration context	Mandatory	Class implem.	
WIPG0238	Pull operation has been abandoned due to enumeration context closure	Mandatory	Class implem.	

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0243	Timeout	Optional	Infrastructure, class implem.	
WIPG0227	Other failure	Optional	Infrastructure, class implem.	

2563

2564 **6.5.14 CloseEnumeration**

2565 **Purpose:**

2566 Close an open enumeration session.

2567 **Operation input parameters:**

2568

Generic Name	Generic Type	Requirement	Description
NamespacePath	NamespacePath	Mandatory	Namespace path of the namespace for the enumeration, as defined in 6.5.10.1 (Context Parameter)
EnumerationContext	EnumerationContext	Mandatory	Enumeration context value, as defined in 6.5.10.2

2569

2570 **Operation output parameters:**

2571 None.

2572 **Description:**

2573 The *CloseEnumeration* operation closes the open enumeration session identified by
2574 *EnumerationContext*.

2575 The enumeration session shall have been established using any of the Open operations.

2576 Enumeration sessions are closed implicitly when exhausted, so this operation only needs to be used
2577 when terminating an enumeration sequence before it is exhausted.

2578 **Preconditions:**

- 2579 • The namespace referenced by *NamespacePath* shall exist. If it does not exist, the operation
2580 shall fail, indicating WIPG0204.
- 2581 • The enumeration session identified by *EnumerationContext* shall be open. If this is not satisfied,
2582 the operation shall fail, indicating WIPG0241.
- 2583 • The value of *EnumerationContext* shall be the enumeration context value returned by the
2584 previous Open or Pull operation on the same enumeration session. If this is not satisfied, the
2585 operation shall fail, indicating WIPG0241.

2586 **Postconditions:**

- 2587
- The enumeration session identified by *EnumerationContext* is closed.
- 2588
- Requirements on ACID properties:
 - Atomicity: Required (related to updates to or deletion of an enumeration context that is maintained by the WBEM server)
 - Update Consistency: N/A
 - Isolation: Required
 - Durability: Required (related to updates to or deletion of an enumeration context that is maintained by the WBEM server)
- 2589
- 2590
- 2591
- 2592
- 2593
- 2594

2595 **Error Messages:**

2596

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM server is shutting down	Optional	Infrastructure	
WIPG0240	WBEM server limits are exceeded	Optional	Infrastructure, class implem.	
WIPG0203	Operation not supported by WBEM server infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure, class implem.	
WIPG0228	Operation not supported by class implementation	Mandatory	Class implem.	
WIPG0241	Invalid enumeration context	Mandatory	Class implem.	
WIPG0239	Pull operation cannot be abandoned	Mandatory	Class implem.	
WIPG0243	Timeout	Optional	Infrastructure, class implem.	
WIPG0227	Other failure	Optional	Infrastructure, class implem.	

2597

2598 **6.5.15 EnumerationCount (deprecated)**2599 **Purpose:**

2600 Estimate the total number of remaining items in an open enumeration session.

2601 **Operation input parameters:**

2602

Generic Name	Generic Type	Requirement	Description
NamespacePath	NamespacePath	Mandatory	Namespace path of the namespace for the enumeration, as defined in 6.5.10.1 (Context Parameter)
EnumerationContext	EnumerationContext	Mandatory	Enumeration context value, as defined in 6.5.10.2

2603

2604 **Operation output parameters:**

2605

Generic Name	Generic Type	Requirement	Description
EnumerationCount	uint64	Mandatory	NULL, or estimated number of remaining items

2606

2607 **Description:**

2608 The *EnumerationCount* operation estimates the total number of remaining items in the open
 2609 enumeration session identified by *EnumerationContext*.

2610 The *EnumerationCount* operation has been deprecated in version 1.1.0 of this document. There is
 2611 no replacement.

2612 The enumeration session shall have been established using any of the Open operations.

2613 If not NULL, the *EnumerationCount* operation output parameter is an estimated count of the number
 2614 of items remaining to be retrieved with subsequent Pull operations. Thus, executing this operation
 2615 immediately after opening the enumeration session provides an estimate of the total number of items
 2616 that will be returned in the enumeration set.

2617 If the WBEM server cannot or will not return an estimated count, it may respond with success and
 2618 the NULL value in the *EnumerationCount* operation output parameter.

2619 This mechanism is intended to assist WBEM clients in determining the overall size of an
 2620 enumeration set and of the number of items remaining in the enumeration session. However,
 2621 because it is an estimate and not an exact number, it should not be used for determining the end of
 2622 an enumeration sequence, i.e., in place of the *EndOfSequence* operation output parameter on Open
 2623 and Pull operations.

2624 **Preconditions:**

- 2625 • The namespace referenced by *NamespacePath* shall exist. If it does not exist, the operation
 2626 shall fail, indicating WIPG0204.
- 2627 • The enumeration session identified by *EnumerationContext* shall be open. If this is not satisfied,
 2628 the operation shall fail, indicating WIPG0241.
- 2629 • The value of *EnumerationContext* shall be the enumeration context value returned by the
 2630 previous Open or Pull operation on the same enumeration session. If this is not satisfied, the
 2631 operation shall fail, indicating WIPG0241.

2632 **Postconditions:**

- 2633 • Requirements on ACID properties:
- 2634 – Atomicity: N/A
- 2635 – Update Consistency: N/A
- 2636 – Isolation: Required
- 2637 – Durability: N/A

2638 **Error Messages:**

2639

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM server is shutting down	Optional	Infrastructure	
WIPG0240	WBEM server limits are exceeded	Optional	Infrastructure, class implem.	
WIPG0203	Operation not supported by WBEM server infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure, class implem.	
WIPG0228	Operation not supported by class implementation	Mandatory	Class implem.	
WIPG0241	Invalid enumeration context	Mandatory	Class implem.	
WIPG0243	Timeout	Optional	Infrastructure, class implem.	
WIPG0227	Other failure	Optional	Infrastructure, class implem.	

2640

2641 **6.6 Method invocation operations**

2642 This subclause defines server operations for the invocation of CIM methods.

2643 **6.6.1 InvokeMethod**2644 **Purpose:**

2645 Invoke a non-static method on an instance.

2646 **Operation input parameters:**

2647

Generic Name	Generic Type	Requirement	Description
InstancePath	InstancePath	Mandatory	Instance path of the instance the method is invoked on (Context Parameter)
MethodName	MethodName	Mandatory	Name of the method being invoked
InParmValues	ParameterValue []	Mandatory	Unordered set of named input parameter values of the method

2648

2649 **Operation output parameters:**

2650

Generic Name	Generic Type	Requirement	Description
OutParmValues	ParameterValue []	Mandatory	Unordered set of named output parameter values of the method
ReturnValue	ReturnValue	Mandatory	Return value of the method

2651

2652 **Description:**

2653 Invoke a CIM method using an instance path. The method may be static or non-static.

2654 Conformant WBEM protocols shall define a mapping for the invocation of CIM methods using an
 2655 instance path, including a mapping of the operation parameters defined in the tables above. These
 2656 rules may map the method invocation to a single operation, map each method to its own separate
 2657 operation, or define any other appropriate mapping.

2658 If the implementation of the method could be invoked, the operation is considered successful,
 2659 regardless of what the semantics of any return values or output parameters is. For example, if a
 2660 method defines that a particular return value indicates an error condition, the method invocation was
 2661 still successful from a perspective of the invocation operation.

2662 **Preconditions:**

- 2663 • The instance referenced by *InstancePath* shall exist. If it does not exist, the operation shall fail,
 2664 indicating WIPG0213.
- 2665 • The creation class of the instance referenced by *InstancePath* shall exist. If it does not exist, the
 2666 operation shall fail, indicating WIPG0214.
- 2667 • The namespace of the instance referenced by *InstancePath* shall exist. If it does not exist, the
 2668 operation shall fail, indicating WIPG0204.
- 2669 • The method to be invoked shall be exposed by the creation class of the instance referenced by
 2670 *InstancePath*. If this is not satisfied, the operation shall fail, indicating WIPG0218.

2671 **Postconditions:**

- 2672 • The CIM method shall have been invoked.
- 2673 • Requirements on ACID properties:
- 2674 – Atomicity: Recommended
- 2675 – Update Consistency: Recommended
- 2676 – Isolation: Recommended
- 2677 – Durability: Required

2678 **Error Messages:**

2679

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM server is shutting down	Optional	Infrastructure	
WIPG0240	WBEM server limits are exceeded	Optional	Infrastructure, class implem.	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0229	Method invocation not supported by WBEM server infrastructure	Mandatory	Infrastructure	
WIPG0218	No such method	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure, class implem.	
WIPG0214	Class not found	Mandatory	Infrastructure	
WIPG0213	Instance not found	Mandatory	Class implem.	
WIPG0219	Method not supported by class implementation	Mandatory	Class implem.	
WIPG0243	Timeout	Optional	Infrastructure, class implem.	
WIPG0227	Other failure	Optional	Infrastructure, class implem.	

2680

2681 **6.6.2 InvokeStaticMethod**2682 **Purpose:**

2683 Invoke a static method on a class.

2684 **Operation input parameters:**
2685

Generic Name	Generic Type	Requirement	Description
ClassPath	ClassPath	Mandatory	Class path of the class the method is invoked on (Context Parameter)
MethodName	MethodName	Mandatory	Name of the method being invoked
InParmValues	ParameterValue []	Mandatory	Unordered set of named input parameter values of the method

2686

2687 **Operation output parameters:**
2688

Generic Name	Generic Type	Requirement	Description
OutParmValues	ParameterValue []	Mandatory	Unordered set of named output parameter values of the method
ReturnValue	ReturnValue	Mandatory	Return value of the method

2689

2690 **Description:**

2691 Invoke a static CIM method using a class path.

2692 Conformant WBEM protocols shall define a mapping for the invocation of CIM methods using a class
2693 path, including a mapping of the operation parameters defined in the tables above. These rules may
2694 map the method invocation to a single operation, map each method to its own separate operation, or
2695 define any other appropriate mapping.

2696 If the implementation of the method could be invoked, the operation is considered successful,
2697 regardless of what the semantics of any return values or output parameters is. For example, if a
2698 method defines that a particular return value indicates an error condition, the method invocation was
2699 still successful from a perspective of the invocation operation.

2700 **Preconditions:**

- 2701 • The class referenced by *ClassPath* shall exist. If it does not exist, the operation shall fail,
2702 indicating WIPG0214.
- 2703 • The namespace of the class referenced by *ClassPath* shall exist. If it does not exist, the
2704 operation shall fail, indicating WIPG0204.
- 2705 • The method to be invoked shall be exposed by the creation class of the instance referenced by
2706 *InstancePath*. If this is not satisfied, the operation shall fail, indicating WIPG0218.

2707 **Postconditions:**

- 2708 • The CIM method shall have been invoked.
- 2709 • Requirements on ACID properties:
 - 2710 – Atomicity: Recommended
 - 2711 – Update Consistency: Recommended

- 2712 – Isolation: Recommended
 2713 – Durability: Required

2714 **Error Messages:**

2715

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM server is shutting down	Optional	Infrastructure	
WIPG0240	WBEM server limits are exceeded	Optional	Infrastructure, class implem.	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0229	Method invocation not supported by WBEM server infrastructure	Mandatory	Infrastructure	
WIPG0218	No such method	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure, class implem.	
WIPG0214	Class not found	Mandatory	Class implem.	
WIPG0219	Method not supported by class implementation	Mandatory	Class implem.	
WIPG0243	Timeout	Optional	Infrastructure, class implem.	
WIPG0227	Other failure	Optional	Infrastructure, class implem.	

2716

2717 **6.7 Class operations**

2718 This subclause defines server operations that target a single class or create a class. These operations
 2719 include dealing with qualifier values defined on classes and their elements.

2720 **6.7.1 GetClass**

2721 **Purpose:**

2722 Retrieve a class.

2723 **Operation input parameters:**

2724

Generic Name	Generic Type	Requirement	Description
ClassPath	ClassPath	Mandatory	Class path of the class to be retrieved (Context Parameter)
IncludeInheritedElements	boolean	Optional	Indicates whether any elements inherited from superclasses are to be included in the returned class
IncludeQualifiers	boolean	Mandatory	Indicates whether qualifier values on any returned CIM elements are to be included, as defined in 6.2.2
IncludeClassOrigin	boolean	Conditional	Indicates whether class origin information for any returned CIM elements within a class is to be included, as defined in 6.2.1 Condition: WBEM protocol supports client side control of returning class origin information.
IncludedProperties	PropertyName []	Optional	Deprecated: NULL, or unordered set of property names to be included, acting as a restricting filter on the properties included in the returned class

2725

2726 **Operation output parameters:**

2727

Generic Name	Generic Type	Requirement	Description
Class	ClassSpecification	Mandatory	Retrieved class representation

2728

2729 **Description:**

2730 The *GetClass* operation retrieves a representation of the class referenced by *ClassPath*.

2731 The set of properties to be included in the retrieved class shall be determined using the following algorithm:

2732

- 2733 • Initially, the set of properties to be included is the set of properties exposed by the class to
2734 be retrieved. This includes all the duplicates of any duplicate non-overridden properties.
- 2735 • **Deprecated:** If the *IncludedProperties* operation input parameter is supported by the
2736 WBEM protocol and if its value is not NULL, it acts as a restricting filter on the properties to
2737 be included in the returned class such that any properties exposed by the class to be
2738 retrieved that are not named in that operation parameter are removed from the set of
2739 properties to be included. Any duplicate or invalid property names in the
2740 *IncludedProperties* operation input parameter shall be ignored. A non-NULL empty
2741 *IncludedProperties* list removes all properties from the set of properties to be included.

2742 The *IncludedProperties* parameter was deprecated in version 1.1.0 of this specification,
2743 with no replacement.

- 2744 • If *IncludeInheritedElements* is FALSE, it acts as a restricting filter on the elements
2745 (properties, methods, qualifiers) to be included in the returned class such that any

2746 elements inherited into the class to be retrieved are removed from the set of properties to
 2747 be included. This is also known as reducing the elements to *local-only* elements.

2748 **Preconditions:**

- 2749 • The class referenced by *ClassPath* shall exist. If it does not exist, the operation shall fail,
 2750 indicating WIPG0214.
- 2751 • The namespace of the class referenced by *ClassPath* shall exist. If it does not exist, the
 2752 operation shall fail, indicating WIPG0204.

2753 **Postconditions:**

- 2754 • The class representation shall have been returned as defined in the Description paragraph for
 2755 this operation.
- 2756 • Requirements on ACID properties:
 - 2757 – Atomicity: N/A
 - 2758 – Update Consistency: N/A
 - 2759 – Isolation: Required
 - 2760 – Durability: N/A

2761 **Error Messages:**

2762

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM server is shutting down	Optional	Infrastructure	
WIPG0240	WBEM server limits are exceeded	Optional	Infrastructure	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM server infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure	
WIPG0214	Class not found	Mandatory	Infrastructure	
WIPG0243	Timeout	Optional	Infrastructure	
WIPG0227	Other failure	Optional	Infrastructure	

2763

2764 **6.7.2 DeleteClass**2765 **Purpose:**

2766 Delete a given class.

2767 **Operation input parameters:**

2768

Generic Name	Generic Type	Requirement	Description
ClassPath	ClassPath	Mandatory	Class path of the class to be deleted (Context Parameter)
DeleteDependents	Boolean	Optional	EXPERIMENTAL: Indicates whether dependent classes and instances are to be deleted as well

2769

2770 **Operation output parameters:**

2771 None.

2772 **Description:**2773 The *DeleteClass* operation deletes the class referenced by *ClassPath*.

2774

2775 EXPERIMENTAL

2776 If the WBEM protocol supports the *DeleteDependents* operation parameter, the following rules apply:

- 2777 • If *DeleteDependents* is TRUE, any classes that depend on the class referenced by
2778 *ClassPath* in the way described below shall be deleted, and any instances of the class
2779 referenced by *ClassPath* and of any classes depending on it shall be deleted according to
2780 the rules defined for the
- 2781 • DeleteInstance operation. If these rules cause the rejection of an instance deletion, the
- 2782 • DeleteClass operation shall fail.
- 2783 • If *DeleteDependents* is FALSE, the *DeleteClass* operation shall fail if any classes exist that
2784 depend on the class referenced by *ClassPath* in the way described below, or if the class
2785 referenced by *ClassPath* has any instances.

2786 EXPERIMENTAL

2787 If the WBEM protocol does not support the *DeleteDependents* operation parameter, the *DeleteClass*
2788 operation shall fail if any classes exist that depend on the class referenced by *ClassPath* in the way
2789 described below, or if the class referenced by *ClassPath* has any instances.

2790 For the purpose of the *DeleteClass* operation, the following classes are considered depending on the
2791 class referenced by *ClassPath*:

- 2792 • Any subclasses of any class depending on the class referenced by *ClassPath*.
- 2793 • Any association classes referencing any class depending on the class referenced by
2794 *ClassPath*.
- 2795 • Any classes defining a method with a parameter or a return value that is
 - 2796 – a reference to any class depending on the class referenced by *ClassPath*, or
 - 2797 – an embedded instance of any class depending on the class referenced by *ClassPath*,
2798 or
 - 2799 – an embedded class depending on the class referenced by *ClassPath*.
- 2800 • Any classes defining a property that is
 - 2801 – an embedded instance of any class depending on the class referenced by *ClassPath*,
2802 or
 - 2803 – an embedded class depending on the class referenced by *ClassPath*.

2804 Any classes or instances that are automatically deleted may reside in a different namespace (which
2805 may reside in a different WBEM server) than the class referenced by *ClassPath*.

2806 In case of error, the consistency requirements defined in [DSP0004](#) cannot be guaranteed, but should
2807 be attempted to be satisfied in a best effort approach. In case of error, only a subset of the elements
2808 to be deleted may have been deleted, but each element shall have either been deleted completely or
2809 not at all. Also, classes shall only be deleted if all of its instances could be deleted successfully.

2810 NOTE In a non-transactional implementation, this requires an order of deletion that starts with those
2811 elements that do not depend on the deletion of other elements.

2812 **Preconditions:**

- 2813 • The class referenced by *ClassPath* shall exist. If it does not exist, the operation shall fail,
2814 indicating WIPG0214.
- 2815 • The namespace of the class referenced by *ClassPath* shall exist. If it does not exist, the
2816 operation shall fail, indicating WIPG0204.

2817 **Postconditions:**

- 2818 • The class referenced by *ClassPath* shall have been deleted.
- 2819 • If *DeleteDependents* was TRUE:
 - 2820 – any dependent classes and instances shall have been deleted as defined in the
2821 Description paragraph for this operation, and
 - 2822 – any management profile defined implicit deletions of other instances shall have
2823 happened, and
 - 2824 – any management profile defined effects of the deletion of all of these instances on any
2825 managed resources shall have happened.
- 2826 • The consistency requirements defined in [DSP0004](#) shall be satisfied for any classes and
2827 instances related to the deleted classes and instances.
- 2828 • Requirements on ACID properties:
 - 2829 – Atomicity: Required, if dependent classes and instances are handled by rejection, as
2830 defined in 5.8.9. Recommended, if dependent classes and instances are handled by
2831 delete propagation, as defined in 5.8.9.
 - 2832 – Update Consistency: Required, if dependent classes and instances are handled by
2833 rejection, as defined in 5.8.9. Recommended, if dependent classes and instances are
2834 handled by delete propagation, as defined in 5.8.9.
 - 2835 – Isolation: Required, if dependent classes and instances are handled by rejection, as
2836 defined in 5.8.9. Recommended, if dependent classes and instances are handled by
2837 delete propagation, as defined in 5.8.9.
 - 2838 – Durability: Required

2839 **Error Messages:**

2840

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM server is shutting down	Optional	Infrastructure	
WIPG0240	WBEM server limits are exceeded	Optional	Infrastructure, class implem.	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM server infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure, class implem.	
WIPG0214	Class not found	Mandatory	Infrastructure	
WIPG0224	Class has subclasses	Mandatory	Infrastructure	
WIPG0225	Class has instances	Mandatory	Infrastructure, class implem.	
WIPG0230	Class has referencing association classes	Mandatory	Infrastructure	
WIPG0243	Timeout	Optional	Infrastructure, class implem.	
WIPG0227	Other failure	Optional	Infrastructure, class implem.	

2841

2842 **6.7.3 ModifyClass**2843 **Purpose:**

2844 Change a given class.

2845 **Operation input parameters:**

2846

Generic Name	Generic Type	Requirement	Description
ClassPath	ClassPath	Mandatory	Class path of the class to be changed. (Context Parameter)
ModifiedClass	ClassSpecification	Mandatory	Class representation specifying the new class definition

2847

2848 **Operation output parameters:**

2849 None.

2850 **Description:**2851 The *ModifyClass* operation changes the definition of the class referenced by *ClassPath*.2852 Within the restrictions specified in the preconditions, the definition of the class referenced by
2853 *ClassPath* is replaced with the definition specified in *ModifiedClass*, as follows:

- 2854 • Any elements previously defined in the class to be changed (including overriding elements)
2855 that are not specified in *ModifiedClass* shall be removed from the class to be changed.
- 2856 • Any elements previously defined in the class to be changed (including overriding elements)
2857 that are also specified in *ModifiedClass* shall be replaced with the definition from
2858 *ModifiedClass*.

- 2859
2860
2861
- Any elements not previously defined in the class to be changed (including overriding elements) that are specified in *ModifiedClass* shall be added to the class to be changed, as defined in *ModifiedClass*.

2862 Any instances whose creation class is the class referenced by *ClassPath* or one of its subclasses
2863 shall be changed to reflect the changes to the class, as follows:

- 2864
2865
2866
- Added properties are reflected using the rules defined in the *ModifyInstance* operation when processing a list of these new properties with their values set to their class defined default values, or NULL where no class defined default value is defined.

2867 Any other changes to the class that are compatible with the preconditions do not affect existing
2868 instances, for the following reasons:

- 2869
2870
2871
2872
- A compatible removal of properties from a class can only happen for overridden properties or for properties that move to a superclass, both of which is equivalent to potential changes of qualifier values and the default property value. Changes of qualifier values do not affect instances. A changed default value only affects new instances, but not existing instances.
- 2873
2874
2875
- A compatible change of existing property definitions can only include potential changes of qualifier values and the default property value. Changes of qualifier values do not affect instances. A changed default value only affects new instances, but not existing instances.
- 2876
- A compatible change of values of class qualifiers does not affect instances of the class.
- 2877
- A compatible change to a method definition does not affect instances of the class.

2878 **Preconditions:**

- 2879
2880
- The class referenced by *ClassPath* shall exist. If it does not exist, the operation shall fail, indicating WIPG0214.
- 2881
2882
- The namespace of the class referenced by *ClassPath* shall exist. If it does not exist, the operation shall fail, indicating WIPG0204.
- 2883
2884
- The name of the class defined by *ModifiedClass* shall be the name of the class referenced by *ClassPath*. If this is not satisfied, the operation shall fail, indicating WIPG0208.
- 2885
2886
2887
2888
- If the class referenced by *ClassPath* has a superclass, the class defined by *ModifiedClass* shall specify a superclass with the same name as that superclass. If the class referenced by *ClassPath* has no superclass, the class defined by *ModifiedClass* shall not specify a superclass. If this is not satisfied, the operation shall fail, indicating WIPG0226.
- 2889
2890
2891
2892
2893
- The class defined by *ModifiedClass* shall only specify elements that when applied to the class to be modified, result in a class definition that satisfies any consistency and backward compatibility requirements defined in [DSP0004](#). For example, qualifiers with flavor *DisableOverride* shall not be overridden, or data types of overridden properties shall not be changed. If this is not satisfied, the operation shall fail, indicating WIPG0231.

2894 **Postconditions:**

- 2895
2896
- The definition of the class referenced by *ClassPath* shall have been modified as defined in the Description paragraph for this operation.
- 2897
2898
- Any instances of the class or its subclasses shall have been changed as defined in the Description paragraph for this operation.
- 2899
2900
- The consistency and backward compatibility requirements defined in [DSP0004](#) shall be satisfied for the modified class.

- 2901 • Requirements on ACID properties:
- 2902 – Atomicity: Required
- 2903 – Update Consistency: Required
- 2904 – Isolation: Required
- 2905 – Durability: Required

2906 **Error Messages:**

2907

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM server is shutting down	Optional	Infrastructure	
WIPG0240	WBEM server limits are exceeded	Optional	Infrastructure	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM server infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure	
WIPG0214	Class not found	Mandatory	Infrastructure	
WIPG0226	Superclass not found	Mandatory	Infrastructure	
WIPG0231	Incompatible class modification	Mandatory	Infrastructure	
WIPG0243	Timeout	Optional	Infrastructure	
WIPG0227	Other failure	Optional	Infrastructure	

2908

2909 **6.7.4 CreateClass**

2910 **Purpose:**

2911 Create a class.

2912 **Operation input parameters:**

2913

Generic Name	Generic Type	Requirement	Description
NamespacePath	NamespacePath	Mandatory	Namespace path of the namespace the class is to be created in (Context Parameter)

Generic Name	Generic Type	Requirement	Description
NewClass	ClassSpecification	Mandatory	Representation of the class to be created

2914

Operation output parameters:

2915

2916

Generic Name	Generic Type	Requirement	Description
ClassPath	ClassPath	Mandatory	Class path of the new class

2917

Description:

2918

2919 The *CreateClass* operation creates a class in the namespace referenced by *NamespacePath*, using
 2920 the class representation in *NewClass*, and returns the class path of the new class.

2921 If properties or methods defined in *NewClass* are intended to override properties or methods defined
 2922 in a superclass of *NewClass*, then they shall define an *OVERRIDE* qualifier in their definition in
 2923 *NewClass*. The *CreateClass* operation shall not add such qualifiers automatically.

Preconditions:

2924

- 2925 • The namespace referenced by *NamespacePath* shall exist. If it does not exist, the operation
 2926 shall fail, indicating WIPG0204.
- 2927 • The class to be created shall not exist in the namespace referenced by *NamespacePath*. If this
 2928 is not satisfied, the operation shall fail, indicating WIPG0217.
- 2929 • If *NewClass* specifies a superclass, that superclass shall exist in the namespace referenced by
 2930 *NamespacePath*. If this is not satisfied, the operation shall fail, indicating WIPG0226.
- 2931 NOTE [DSP0004](#) does not provide for inheritance relationships that cross namespace boundaries.
- 2932 • The definition of *NewClass* shall satisfy any consistency requirements defined in [DSP0004](#). If
 2933 this is not satisfied, the operation shall fail, indicating WIPG0208.

Postconditions:

2934

- 2935 • The class shall have been created as defined in the Description paragraph for this operation.
- 2936 • Requirements on ACID properties:
 - 2937 – Atomicity: Required
 - 2938 – Update Consistency: Required
 - 2939 – Isolation: Required
 - 2940 – Durability: Required

Error Messages:

2941

2942

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM server is shutting down	Optional	Infrastructure	

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0240	WBEM server limits are exceeded	Optional	Infrastructure	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM server infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure	
WIPG0217	Class already exists	Mandatory	Infrastructure	
WIPG0226	Superclass not found	Mandatory	Infrastructure	
WIPG0243	Timeout	Optional	Infrastructure	
WIPG0227	Other failure	Optional	Infrastructure	

2943

2944 **6.8 Class enumeration operations**

2945 This subclause defines server operations that enumerate classes and return their representations and
 2946 class paths.

2947 **6.8.1 EnumerateClasses**2948 **Purpose:**

2949 Enumerate classes in a namespace and return these classes together with their class paths.

2950 **Operation input parameters:**

2951

Generic Name	Generic Type	Requirement	Description
NamespacePath	NamespacePath	Mandatory	Namespace path of the namespace the enumeration is executed on (Context Parameter)
ClassName	ClassName	Mandatory	Optional: Name of the CIM class whose subclasses are to be enumerated. If not specified, top classes are enumerated.
IncludeSubclasses	boolean	Mandatory	Indicates whether the entire tree of subclasses is to be included in the result set, in addition
IncludeInheritedElements	boolean	Mandatory	Indicates whether any elements inherited from superclasses of ClassName are to be included in the returned classes

Generic Name	Generic Type	Requirement	Description
IncludeQualifiers	boolean	Mandatory	Indicates whether qualifier values on any returned CIM elements are to be included, as defined in 6.2.2
IncludeClassOrigin	boolean	Conditional	Indicates whether class origin information for any returned CIM elements within a class is to be included, as defined in 6.2.1 Condition: WBEM protocol supports client side control of returning class origin information

2952

2953 **Operation output parameters:**

2954

Generic Name	Generic Type	Requirement	Description
ClassList	ClassSpecificationWithPath []	Mandatory	Sequence of the enumerated classes with their class paths

2955

2956 **Description:**

2957 The *EnumerateClasses* operation enumerates classes (including association and indication classes)
2958 in the namespace specified in *NamespacePath* and returns their representations and class paths.

2959 *ClassName* and *IncludeSubclasses* together determine the set of classes in the result set. The set of
2960 classes in the result set is determined using the following algorithm:

- 2961 1) *ClassName* is optional to be specified by the WBEM client (Note that *ClassName* is
2962 mandatory to be supported by the WBEM protocol). If *ClassName* is not specified, the
2963 result set initially contains all top classes (that is, classes that do not have a superclass) in
2964 the namespace. If *ClassName* is specified, the result set initially contains the subclasses of
2965 the class specified in *ClassName* (not including the class specified in *ClassName*).
- 2966 2) If *IncludeSubclasses* is TRUE, then all direct and indirect subclasses of the classes that
2967 are so far in the result set are added to the result set. Otherwise, the result set is not
2968 changed.

2969 If *IncludeInheritedElements* is TRUE, then the set of CIM elements in each returned class shall
2970 consist of all elements exposed by that class. Otherwise, the set of CIM elements in each returned
2971 class shall consist only of all elements defined in the class specified in *ClassName* (including
2972 overriding elements).

2973 The consistency model defined in 5.8 applies.

2974 **Preconditions:**

- 2975 • The namespace referenced by *NamespacePath* shall exist. If it does not exist, the operation
2976 shall fail, indicating WIPG0204.
- 2977 • If *ClassName* is specified, the specified CIM class shall exist in the namespace referenced by
2978 *NamespacePath*. If this is not satisfied, the operation shall fail, indicating WIPG0214.

2979 **Postconditions:**

- 2980 • The enumerated classes with their class paths shall have been returned as defined in the
2981 Description paragraph for this operation.
- 2982 • Requirements on ACID properties:
- 2983 – Atomicity: N/A
- 2984 – Update Consistency: N/A
- 2985 – Isolation: Required at the level of single classes, as defined in 5.8.
- 2986 – Durability: N/A

2987 **Error Messages:**

2988

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM server is shutting down	Optional	Infrastructure	
WIPG0240	WBEM server limits are exceeded	Optional	Infrastructure	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM server infrastructure	Mandatory	Infrastructure	
WIPG0214	Class not found	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure	
WIPG0243	Timeout	Optional	Infrastructure	
WIPG0227	Other failure	Optional	Infrastructure	

2989

2990 **6.8.2 EnumerateClassNames**

2991 **Purpose:**

2992 Enumerate classes in a namespace and return their class names.

2993 **Operation input parameters:**

2994

Generic Name	Generic Type	Requirement	Description
NamespacePath	NamespacePath	Mandatory	Namespace path of the CIM namespace the enumeration is executed on (Context Parameter)
ClassName	ClassName	Mandatory	Optional: Name of the CIM class whose subclasses are to be enumerated. If not specified, top classes are enumerated.
IncludeSubclasses	boolean	Mandatory	Indicates whether the entire tree of subclasses is to be included in the result set, in addition

2995

2996 **Operation output parameters:**

2997

Generic Name	Generic Type	Requirement	Description
ClassNameList	ClassName []	Mandatory	Sequence of class names of the enumerated classes

2998

2999 **Description:**

3000 The *EnumerateClassNames* operation enumerates classes (including association and indication
 3001 classes) in the namespace specified in *NamespacePath* and returns the class names of these
 3002 classes (that is, just the class name, not the class path).

3003 *ClassName* and *IncludeSubclasses* together determine the set of class names in the result set. The
 3004 set of class names in the result set is determined using the following algorithm:

- 3005 1) *ClassName* is optional to be specified by the WBEM client (Note that *ClassName* is
 3006 mandatory to be supported by the WBEM protocol). If *ClassName* is not specified, the
 3007 result set initially contains the names of all top classes (that is, classes that do not have a
 3008 superclass) in the namespace. If *ClassName* is specified, the result set initially contains the
 3009 names of the subclasses of the class specified in *ClassName* (not including the class name
 3010 specified in *ClassName*).
- 3011 2) If *IncludeSubclasses* is True, the class names of all direct and indirect subclasses of the
 3012 classes that are so far in the result set are added to the result set. Otherwise, the result set
 3013 is not changed.

3014 The consistency model defined in 5.8 applies.

3015 **Preconditions:**

- 3016 • The namespace referenced by *NamespacePath* shall exist. If it does not exist, the operation
3017 shall fail, indicating WIPG0204.
- 3018 • If *ClassName* is specified, the specified CIM class shall exist in the namespace referenced by
3019 *NamespacePath*. If this is not satisfied, the operation shall fail, indicating WIPG0214.

3020 **Postconditions:**

- 3021 • The class names of the enumerated classes shall have been returned as defined in the
3022 Description paragraph for this operation.
- 3023 • Requirements on ACID properties:
- 3024 – Atomicity: N/A
- 3025 – Update Consistency: N/A
- 3026 – Isolation: Required at the level of single classes, as defined in 5.8.
- 3027 – Durability: N/A

3028 **Error Messages:**

3029

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM server is shutting down	Optional	Infrastructure	
WIPG0240	WBEM server limits are exceeded	Optional	Infrastructure	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM server infrastructure	Mandatory	Infrastructure	
WIPG0214	Class not found	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure	
WIPG0243	Timeout	Optional	Infrastructure	
WIPG0227	Other failure	Optional	Infrastructure	

3030

3031 **6.8.3 AssociatorClasses**

3032 **Purpose:**

3033 Enumerate the classes that are associated with a given source class and return their class
 3034 representations and class paths.

3035 **Operation input parameters:**

3036

Generic Name	Generic Type	Requirement	Description
ClassPath	ClassPath	Mandatory	Class path of the source class (Context Parameter)
AssociationClassName	ClassName	Mandatory	NULL, or name of the association class, acting as a restricting filter on the associated classes
AssociatedClassName	ClassName	Mandatory	NULL, or name of the associated class on any far end of the association, acting as a restricting filter on the associated classes
SourceRoleName	PropertyName	Mandatory	NULL, or name of the role on the starting end of the association, acting as a restricting filter on the associated classes
AssociatedRoleName	PropertyName	Mandatory	NULL, or name of the role on any far end of the association, acting as a restricting filter on the associated classes
IncludeQualifiers	boolean	Mandatory	Indicates whether qualifier values on any returned CIM elements are to be included, as defined in 6.2.2
IncludeClassOrigin	boolean	Conditional	Indicates whether class origin information for any returned CIM elements within a class is to be included, as defined in 6.2.1 Condition: WBEM protocol supports client side control of returning class origin information.
IncludedProperties	PropertyName []	Optional	Deprecated: NULL, or unordered set of property names to be included, acting as a restricting filter on the properties included in the returned class

3037

3038 **Operation output parameters:**

3039

Generic Name	Generic Type	Requirement	Description
ClassList	ClassSpecificationWithPath []	Mandatory	Sequence of the returned class representations and class paths

3040

3041 **Description:**

3042 The *AssociatorClasses* operation traverses an association from a given source class on a starting
 3043 end to classes on all of its far ends and returns the associated classes together with their class
 3044 paths.

- 3045 The set of associated classes to be enumerated shall be determined using the following algorithm:
- 3046
- 3047
- Initially, the set of classes to be enumerated is the set of all classes associated to any of the far ends of all associations referencing the starting class.
 - If the *AssociationClassName* operation input parameter is not NULL, it acts as a restricting filter on the classes to be enumerated such that each class that is associated with the starting class using an association class where the class or one of its superclasses does not have the name specified in *AssociationClassName*, is removed from the set of classes to be enumerated. There shall be no validity checking performed for the *AssociationClassName* operation input parameter; if the specified class does not exist, the operation shall succeed with an empty result (because the filter did not match).
 - If the *AssociatedClassName* operation input parameter is not NULL, it acts as a restricting filter on the classes to be enumerated such that each class where the class or one of its superclasses does not have the name specified in *AssociatedClassName*, is removed from the set of classes to be enumerated. There shall be no validity checking performed for the *AssociatedClassName* operation input parameter; if the specified class does not exist, the operation shall succeed with an empty result (because the filter did not match).
- 3055
- 3056
- 3057
- 3058
- 3059
- 3060
- 3061 NOTE Specifying a non-NULL value for *AssociatedClassName* ensures that the returned classes
- 3062 have the class specified in *AssociatedClassName* as a common superclass.
- If the *SourceRoleName* operation input parameter is not NULL, it acts as a restricting filter on the classes to be enumerated such that each class that is associated with the starting class using an association class that has a role name on its starting end that is not the role name specified in *SourceRoleName*, is removed from the set of classes to be enumerated. There shall be no validity checking performed for the *SourceRoleName* operation input parameter; if the specified role does not exist, the operation shall succeed with an empty result (because the filter did not match).
 - If the *AssociatedRoleName* operation input parameter is not NULL, it acts as a restricting filter on the classes to be enumerated such that each class that is associated with the starting class using an association class that has a role name on the far end referencing that class that is not the role name specified in *AssociatedRoleName*, is removed from the set of classes to be enumerated. There shall be no validity checking performed for the *AssociatedRoleName* operation input parameter; if the specified role does not exist, the operation shall succeed with an empty result (because the filter did not match).
- 3070
- 3071
- 3072
- 3073
- 3074
- 3075
- 3076
- 3077 The consistency model defined in 5.8 applies.
- 3078 The set of properties to be included in each returned associated class shall be determined using the
- 3079 following algorithm:
- Initially, the set of properties to be included is the set of properties exposed by the class. This includes all the duplicates of any duplicate non-overridden properties.
 - **Deprecated:** If the *IncludedProperties* operation input parameter is supported by the WBEM protocol and if its value is not NULL, it acts as a restricting filter on the properties to be included in the returned classes such that any properties exposed by the associated class that are not named in that operation parameter are removed from the set of properties to be included. Any duplicate or invalid property names in the *IncludedProperties* operation input parameter shall be ignored. A non-NULL empty *IncludedProperties* list removes all properties from the set of properties to be included.
- 3089 The *IncludedProperties* parameter was deprecated in version 1.1.0 of this specification,
- 3090 with no replacement.

3091 **Preconditions:**

- 3092 • The namespace of the source class referenced by *ClassPath* shall exist. If it does not exist, the
3093 operation shall fail, indicating WIPG0204.
- 3094 • **Deprecated:** The *IncludedProperties* operation parameter, if supported by the WBEM protocol,
3095 shall only be specified with a non-NULL value if the *AssociatedClassName* operation input
3096 parameter is also non-NULL. If this is not satisfied, the operation shall fail, indicating
3097 WIPG0208.
- 3098 • The namespace of any returned classes shall exist. If it does not exist, the operation shall fail,
3099 indicating WIPG0204. Note that cross-namespace association traversals may return classes in
3100 a server or namespace that is different from the server or namespace of the source class.

3101 **Postconditions:**

- 3102 • The associated classes with their class paths shall have been returned as described in the
3103 Description paragraph for this operation.
- 3104 • Requirements on ACID properties:
 - 3105 – Atomicity: N/A
 - 3106 – Update Consistency: N/A
 - 3107 – Isolation: Required at the level of single classes, as defined in 5.8.
 - 3108 – Durability: N/A

3109 **Error Messages:**

3110

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM server is shutting down	Optional	Infrastructure	
WIPG0240	WBEM server limits are exceeded	Optional	Infrastructure	
WIPG0204	Namespace not found	Mandatory	Infrastructure	For input namespace and namespace of returned class paths
WIPG0203	Operation not supported by WBEM server infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure	
WIPG0243	Timeout	Optional	Infrastructure	
WIPG0227	Other failure	Optional	Infrastructure	

3111

3112 NOTE Version 1.1 of this specification removed WIPG0214 (Class not found) from this table.

3113 **6.8.4 AssociatorClassPaths**3114 **Purpose:**

3115 Enumerate the classes that are associated with a given source class and return their class paths.

3116 **Operation input parameters:**

3117

Generic Name	Generic Type	Requirement	Description
ClassPath	ClassPath	Mandatory	Class path of the source class (Context Parameter)
AssociationClassName	ClassName	Mandatory	NULL, or name of the association class, acting as a restricting filter on the associated classes
AssociatedClassName	ClassName	Mandatory	NULL, or name of the associated class on any far end of the association, acting as a restricting filter on the associated classes
SourceRoleName	PropertyName	Mandatory	NULL, or name of the role on the starting end of the association, acting as a restricting filter on the associated classes
AssociatedRoleName	PropertyName	Mandatory	NULL, or name of the role on any far end of the association, acting as a restricting filter on the associated classes

3118

3119 **Operation output parameters:**

3120

Generic Name	Generic Type	Requirement	Description
ClassPathList	ClassPath []	Mandatory	Sequence of the class paths of the associated classes

3121

3122 **Description:**3123 The *AssociatorClassPaths* operation traverses an association from a class on a starting end to
3124 classes on all of its far ends and returns the class paths of the associated classes.

3125 The set of associated classes to be enumerated shall be determined using the following algorithm:

- 3126 • Initially, the set of classes to be enumerated is the set of all classes associated to any of
3127 the far ends of all associations referencing the starting class.
- 3128 • If the *AssociationClassName* operation input parameter is not NULL, it acts as a restricting
3129 filter on the classes to be enumerated such that each class that is associated with the
3130 starting class using an association class where the class or one of its superclasses does
3131 not have the name specified in *AssociationClassName*, is removed from the set of classes
3132 to be enumerated. There shall be no validity checking performed for the
3133 *AssociationClassName* operation input parameter; if the specified class does not exist, the
3134 operation shall succeed with an empty result (because the filter did not match).
- 3135 • If the *AssociatedClassName* operation input parameter is not NULL, it acts as a restricting
3136 filter on the classes to be enumerated such that each class where the class or one of its
3137 superclasses does not have the name specified in *AssociatedClassName*, is removed from
3138 the set of classes to be enumerated. There shall be no validity checking performed for the

3139 *AssociatedClassName* operation input parameter; if the specified class does not exist, the
 3140 operation shall succeed with an empty result (because the filter did not match).

3141 NOTE Specifying a non-NULL value for *AssociatedClassName* ensures that the returned classes
 3142 have the class specified in *AssociatedClassName* as a common superclass.

3143 • If the *SourceRoleName* operation input parameter is not NULL, it acts as a restricting filter
 3144 on the classes to be enumerated such that each class that is associated with the starting
 3145 class using an association class that has a role name on its starting end that is not the role
 3146 name specified in *SourceRoleName*, is removed from the set of classes to be enumerated.
 3147 There shall be no validity checking performed for the *SourceRoleName* operation input
 3148 parameter; if the specified role does not exist, the operation shall succeed with an empty
 3149 result (because the filter did not match).

3150 • If the *AssociatedRoleName* operation input parameter is not NULL, it acts as a restricting
 3151 filter on the classes to be enumerated such that each class that is associated with the
 3152 starting class using an association class that has a role name on the far end referencing
 3153 that class that is not the role name specified in *AssociatedRoleName*, is removed from the
 3154 set of classes to be enumerated. There shall be no validity checking performed for the
 3155 *AssociatedRoleName* operation input parameter; if the specified role does not exist, the
 3156 operation shall succeed with an empty result (because the filter did not match).

3157 The consistency model defined in 5.8 applies.

3158 **Preconditions:**

- 3159 • The namespace of the source class referenced by *ClassPath* shall exist. If it does not exist, the
 3160 operation shall fail, indicating WIPG0204.
- 3161 • The namespace of any returned classes shall exist. If it does not exist, the operation shall fail,
 3162 indicating WIPG0204. Note that cross-namespace association traversals may return classes in
 3163 a server or namespace that is different from the server or namespace of the source class.

3164 **Postconditions:**

- 3165 • The class paths of the associated classes shall have been returned as described in the
 3166 Description paragraph for this operation.
- 3167 • Requirements on ACID properties:
 - 3168 – Atomicity: N/A
 - 3169 – Update Consistency: N/A
 - 3170 – Isolation: Required at the level of single classes, as defined in 5.8.
 - 3171 – Durability: N/A

3172 **Error Messages:**

3173

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM server is shutting down	Optional	Infrastructure	
WIPG0240	WBEM server limits are exceeded	Optional	Infrastructure	

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0204	Namespace not found	Mandatory	Infrastructure	For input namespace and namespace of returned class paths
WIPG0203	Operation not supported by WBEM server infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure	
WIPG0243	Timeout	Optional	Infrastructure	
WIPG0227	Other failure	Optional	Infrastructure	

3174

3175 NOTE Version 1.1 of this specification removed WIPG0214 (Class not found) from this table.

3176 **6.8.5 ReferenceClasses**3177 **Purpose:**

3178 Enumerate the association classes that reference a given source class and return their
 3179 representations and class paths.

3180 **Operation input parameters:**

3181

Generic Name	Generic Type	Requirement	Description
ClassPath	ClassPath	Mandatory	Class path of the source class (Context Parameter)
AssociationClassName	ClassName	Mandatory	NULL, or name of the association class, acting as a restricting filter on the association classes
SourceRoleName	PropertyName	Mandatory	NULL, or name of the role on the starting end of the association, acting as a restricting filter on the association classes
IncludeQualifiers	boolean	Mandatory	Indicates whether qualifier values on any returned CIM elements are to be included, as defined in 6.2.2
IncludeClassOrigin	boolean	Conditional	Indicates whether class origin information for any returned CIM elements within a class is to be included, as defined in 6.2.1 Condition: WBEM protocol supports client side control of returning class origin information.
IncludedProperties	PropertyName []	Optional	Deprecated: NULL, or unordered set of property names to be included, acting as a restricting filter on the properties included in the returned classes

3182

3183 **Operation output parameters:**

3184

Generic Name	Generic Type	Requirement	Description
ClassList	ClassSpecificationWithPath []	Mandatory	Sequence of the CIM association classes

3185

3186 **Description:**

3187 The *ReferenceClasses* operation traverses an association from a class on a starting end to classes
 3188 on all of its far ends and returns the CIM association classes traversed together with their class
 3189 paths.

3190 The set of association classes to be enumerated shall be determined using the following algorithm:

- 3191 • Initially, the set of classes to be enumerated is the set of all association classes referencing
 3192 the starting class.
- 3193 • If the *AssociationClassName* operation input parameter is not NULL, it acts as a restricting
 3194 filter on the classes to be enumerated such that each association class where the class or
 3195 one of its superclasses does not have the name specified in *AssociationClassName*, is
 3196 removed from the set of classes to be enumerated. There shall be no validity checking
 3197 performed for the *AssociationClassName* operation input parameter; if the specified class
 3198 does not exist, the operation shall succeed with an empty result (because the filter did not
 3199 match).

3200 NOTE Specifying a non-NULL value for *AssociationClassName* ensures that the returned classes
 3201 have the class specified in *AssociationClassName* as a common superclass.

- 3202 • If the *SourceRoleName* operation input parameter is not NULL, it acts as a restricting filter
 3203 on the classes to be enumerated such that each association class that has a role name on
 3204 its starting end that is not the role name specified in *SourceRoleName*, is removed from
 3205 the set of classes to be enumerated. There shall be no validity checking performed for the
 3206 *SourceRoleName* operation input parameter; if the specified role does not exist, the
 3207 operation shall succeed with an empty result (because the filter did not match).

3208 NOTE Version 1.1 of this specification removed the *AssociatedClassName* and *AssociatedRoleName* filters
 3209 from this operation.

3210 The consistency model defined in 5.8 applies.

3211 The set of properties to be included in each returned association class shall be determined using the
 3212 following algorithm:

- 3213 • Initially, the set of properties to be included is the set of properties exposed by the
 3214 association class. This includes all the duplicates of any duplicate non-overridden
 3215 properties.
- 3216 • **Deprecated:** If the *IncludedProperties* operation input parameter is supported by the
 3217 WBEM protocol and if its value is not NULL, it acts as a restricting filter on the properties to
 3218 be included in the returned classes such that any properties exposed by the associated
 3219 class that are not named in that operation parameter are removed from the set of
 3220 properties to be included. Any duplicate or invalid property names in the
 3221 *IncludedProperties* operation input parameter shall be ignored. A non-NULL empty
 3222 *IncludedProperties* list removes all properties from the set of properties to be included.

3223 The *IncludedProperties* parameter was deprecated in version 1.1.0 of this specification,
 3224 with no replacement.

3225 **Preconditions:**

- 3226
- 3227
- The namespace of the source class referenced by *ClassPath* shall exist. If it does not exist, the operation shall fail, indicating WIPG0204.
- 3228
- 3229
- 3230
- 3231
- **Deprecated:** The IncludedProperties operation parameter, if supported by the WBEM protocol, shall only be specified with a non-NULL value if the AssociationClassName operation input parameter is also non-NULL. If this is not satisfied, the operation shall fail, indicating WIPG0208.
- 3232
- 3233
- 3234
- The namespace of any returned classes shall exist. If it does not exist, the operation shall fail, indicating WIPG0204. Note that cross-namespace association traversals may return classes in a server or namespace that is different from the server or namespace of the source class.

3235 **Postconditions:**

- 3236
- 3237
- The association classes with their class paths shall have been returned as described in the Description paragraph for this operation.
- 3238
- Requirements on ACID properties:
 - Atomicity: N/A
 - Update Consistency: N/A
 - Isolation: Required at the level of single classes, as defined in 5.8.
 - Durability: N/A
- 3239
- 3240
- 3241
- 3242

3243 **Error Messages:**

3244

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM server is shutting down	Optional	Infrastructure	
WIPG0240	WBEM server limits are exceeded	Optional	Infrastructure	
WIPG0204	Namespace not found	Mandatory	Infrastructure	For input namespace and namespace of returned class paths
WIPG0203	Operation not supported by WBEM server infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure	
WIPG0243	Timeout	Optional	Infrastructure	
WIPG0227	Other failure	Optional	Infrastructure	

3245

3246 NOTE Version 1.1 of this specification removed WIPG0214 (Class not found) from this table.

3247 **6.8.6 ReferenceClassPaths**

3248 **Purpose:**

3249 Enumerate the association classes that reference a given source class and return their class paths.

3250 **Operation input parameters:**

3251

Generic Name	Generic Type	Requirement	Description
ClassPath	ClassPath	Mandatory	Class path of the source class (Context Parameter)
AssociationClassName	ClassName	Mandatory	NULL, or name of the association class, acting as a restricting filter on the association classes
SourceRoleName	PropertyName	Mandatory	NULL, or name of the role on the starting end of the association, acting as a restricting filter on the association classes

3252

3253 **Operation output parameters:**

3254

Generic Name	Generic Type	Requirement	Description
ClassPathList	ClassPath []	Mandatory	Sequence of class paths of the association classes

3255

3256 **Description:**

3257 The *ReferenceClassPaths* operation traverses an association from a class on a starting end to
 3258 classes on all of its far ends and returns the class paths of the association classes traversed.

3259 The set of association classes to be enumerated shall be determined using the following algorithm:

- 3260 • Initially, the set of classes to be enumerated is the set of all association classes referencing
 3261 the starting class.
- 3262 • If the *AssociationClassName* operation input parameter is not NULL, it acts as a restricting
 3263 filter on the classes to be enumerated such that each association class where the class or
 3264 one of its superclasses does not have the name specified in *AssociationClassName*, is
 3265 removed from the set of classes to be enumerated. There shall be no validity checking
 3266 performed for the *AssociationClassName* operation input parameter; if the specified class
 3267 does not exist, the operation shall succeed with an empty result (because the filter did not
 3268 match).

3269 NOTE Specifying a non-NULL value for *AssociationClassName* ensures that the returned classes
 3270 have the class specified in *AssociationClassName* as a common superclass.

- 3271 • If the *SourceRoleName* operation input parameter is not NULL, it acts as a restricting filter
 3272 on the classes to be enumerated such that each association class that has a role name on
 3273 its starting end that is not the role name specified in *SourceRoleName*, is removed from
 3274 the set of classes to be enumerated. There shall be no validity checking performed for the
 3275 *SourceRoleName* operation input parameter; if the specified role does not exist, the
 3276 operation shall succeed with an empty result (because the filter did not match).

3277 NOTE Version 1.1 of this specification removed the *AssociatedClassName* and *AssociatedRoleName* filters
 3278 from this operation.

3279 The consistency model defined in 5.8 applies.

3280 **Preconditions:**

- 3281 • The namespace of the source class referenced by *ClassPath* shall exist. If it does not exist, the
3282 operation shall fail, indicating WIPG0204.
- 3283 • The namespace of any returned classes shall exist. If it does not exist, the operation shall fail,
3284 indicating WIPG0204. Note that cross-namespace association traversals may return classes in
3285 a server or namespace that is different from the server or namespace of the source class.

3286 **Postconditions:**

- 3287 • The association classes with their class paths shall have been returned as described in the
3288 Description paragraph for this operation.
- 3289 • Requirements on ACID properties:
 - 3290 – Atomicity: N/A
 - 3291 – Update Consistency: N/A
 - 3292 – Isolation: Required at the level of single classes, as defined in 5.8.
 - 3293 – Durability: N/A

3294 **Error Messages:**

3295

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM server is shutting down	Optional	Infrastructure	
WIPG0240	WBEM server limits are exceeded	Optional	Infrastructure	
WIPG0204	Namespace not found	Mandatory	Infrastructure	For input namespace and namespace of returned class paths
WIPG0203	Operation not supported by WBEM server infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure	
WIPG0243	Timeout	Optional	Infrastructure	
WIPG0227	Other failure	Optional	Infrastructure	

3296

3297 **NOTE** Version 1.1 of this specification removed WIPG0214 (Class not found) from this table.

3298 **6.9 Qualifier type operations**

3299 This subclause defines server operations that deal with qualifier types. As defined in [DSP0004](#), qualifier
 3300 types represent the declarations of qualifiers, not their values.

3301 **6.9.1 GetQualifierType**

3302 **Purpose:**

3303 Retrieve a qualifier type.

3304 **Operation input parameters:**

3305

Generic Name	Generic Type	Requirement	Description
QualifierTypePath	QualifierTypePath	Mandatory	Qualifier type path of the qualifier type to be retrieved (Context Parameter)

3306

3307 **Operation output parameters:**

3308

Generic Name	Generic Type	Requirement	Description
QualifierType	QualifierType	Mandatory	Representation of the returned qualifier type

3309

3310 **Description:**

3311 The *GetQualifierType* operation retrieves a representation of the qualifier type referenced by
 3312 *QualifierTypePath*.

3313 **Preconditions:**

- 3314 • The qualifier type referenced by *QualifierTypePath* shall exist. If it does not exist, the operation
 3315 shall fail, indicating WIPG0215.
- 3316 • The namespace of the qualifier type referenced by *QualifierTypePath* shall exist. If it does not
 3317 exist, the operation shall fail, indicating WIPG0204.

3318 **Postconditions:**

- 3319 • The representation of the qualifier type shall have been returned as described in the Description
 3320 paragraph for this operation.
- 3321 • Requirements on ACID properties:
 - 3322 – Atomicity: N/A
 - 3323 – Update Consistency: N/A
 - 3324 – Isolation: Required
 - 3325 – Durability: N/A

3326 **Error Messages:**
3327

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM server is shutting down	Optional	Infrastructure	
WIPG0240	WBEM server limits are exceeded	Optional	Infrastructure	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM server infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure	
WIPG0215	Qualifier type not found	Mandatory	Infrastructure	
WIPG0243	Timeout	Optional	Infrastructure	
WIPG0227	Other failure	Optional	Infrastructure	

3328

3329 **6.9.2 DeleteQualifierType**3330 **Purpose:**

3331 Delete a given qualifier type.

3332 **Operation input parameters:**

3333

Generic Name	Generic Type	Requirement	Description
QualifierTypePath	QualifierTypePath	Mandatory	Qualifier type path of the qualifier type to be deleted (Context Parameter)

3334

3335 **Operation output parameters:**

3336 None.

3337 **Description:**3338 The *DeleteQualifierType* operation deletes the qualifier type referenced by *QualifierTypePath*.

3339 As defined in [DSP0004](#), any namespace needs to contain qualifier types for the meta qualifiers and
3340 standard qualifiers, and may contain qualifier types for the optional qualifiers. Thus, deleting any
3341 required qualifier types from a namespace will render that namespace non-compliant to [DSP0004](#).

3342 **Preconditions:**

- 3343 • The qualifier type referenced by *QualifierTypePath* shall exist. If it does not exist, the operation
3344 shall fail, indicating WIPG0215.
- 3345 • The namespace of the qualifier type referenced by *QualifierTypePath* shall exist. If it does not
3346 exist, the operation shall fail, indicating WIPG0204.
- 3347 • The qualifier identified by *QualifierTypePath* shall not be specified on any element in the same
3348 namespace. If this is not satisfied, the operation shall fail, indicating WIPG0233.

3349 **Postconditions:**

- 3350 • The qualifier type shall have been deleted as described in the Description paragraph for this
3351 operation.
- 3352 • Requirements on ACID properties:
 - 3353 – Atomicity: Required
 - 3354 – Update Consistency: Required
 - 3355 – Isolation: Required
 - 3356 – Durability: Required

3357 **Error Messages:**

3358

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM server is shutting down	Optional	Infrastructure	
WIPG0240	WBEM server limits are exceeded	Optional	Infrastructure	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM server infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure	
WIPG0215	Qualifier type not found	Mandatory	Infrastructure	
WIPG0233	Qualifier type is used	Mandatory	Infrastructure	
WIPG0243	Timeout	Optional	Infrastructure	
WIPG0227	Other failure	Optional	Infrastructure	

3359

3360 **6.9.3 ModifyQualifierType**3361 **Purpose:**

3362 Change a given qualifier type.

3363 **Operation input parameters:**

3364

Generic Name	Generic Type	Requirement	Description
QualifierTypePath	QualifierTypePath	Mandatory	Qualifier type path of the qualifier type to be changed (Context Parameter)
ModifiedQualifierType	QualifierType	Mandatory	Representation of the changed qualifier type

3365

3366 **Operation output parameters:**

3367 None.

3368 **Description:**3369 The *ModifyQualifierType* operation changes the qualifier type referenced by *QualifierTypePath*.3370 The qualifier type referenced by *QualifierTypePath* is replaced with the qualifier type representation
3371 specified in *ModifiedQualifierType*.3372 As defined in [DSP0004](#), any namespace needs to contain qualifier types for the meta qualifiers and
3373 standard qualifiers, and may contain qualifier types for the optional qualifiers. Thus, changing these
3374 qualifier types in a namespace inconsistently with their [DSP0004](#) definition will render that
3375 namespace non-compliant to [DSP0004](#).3376 **Preconditions:**3377 • The qualifier type referenced by *QualifierTypePath* shall exist. If it does not exist, the operation
3378 shall fail, indicating WIPG0215.3379 • The namespace of the qualifier type referenced by *QualifierTypePath* shall exist. If it does not
3380 exist, the operation shall fail, indicating WIPG0204.3381 • The name of the qualifier type representation specified in *ModifiedQualifierType* shall equal the
3382 name of the qualifier type referenced by *QualifierTypePath*. If this is not satisfied, the operation
3383 shall fail, indicating WIPG0208.3384 • The request to modify the qualifier type shall satisfy any backward compatibility requirements
3385 defined in [DSP0004](#). If this is not satisfied, the operation shall fail, indicating WIPG0234.3386 • If the qualifier type referenced by *QualifierTypePath* is one of the qualifiers defined in [DSP0004](#),
3387 (i.e., meta, standard, and optional qualifiers), the new definition of the qualifier in
3388 *ModifiedQualifierType* shall be consistent with the definition of the qualifier in [DSP0004](#). If this is
3389 not satisfied, the operation shall fail, indicating WIPG0245.

3390 **Postconditions:**

- 3391 • The qualifier type referenced by *QualifierTypePath* shall have been modified as defined in the
3392 Description paragraph for this operation.
- 3393 • The backward compatibility requirements defined in [DSP0004](#) shall be satisfied for the modified
3394 qualifier type.
- 3395 • Requirements on ACID properties:
 - 3396 – Atomicity: Required
 - 3397 – Update Consistency: Required
 - 3398 – Isolation: Required
 - 3399 – Durability: Required

3400 **Error Messages:**

3401

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM server is shutting down	Optional	Infrastructure	
WIPG0240	WBEM server limits are exceeded	Optional	Infrastructure	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM server infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure	
WIPG0215	Qualifier type not found	Mandatory	Infrastructure	
WIPG0234	Incompatible modification of qualifier type	Mandatory	Infrastructure	
WIPG0245	Qualifier type inconsistent with DSP0004	Mandatory	Infrastructure	
WIPG0243	Timeout	Optional	Infrastructure	
WIPG0227	Other failure	Optional	Infrastructure	

3402

3403 **6.9.4 CreateQualifierType**

3404 **Purpose:**

3405 Create a qualifier type.

3406 **Operation input parameters:**

3407

Generic Name	Generic Type	Requirement	Description
NamespacePath	NamespacePath	Mandatory	Namespace path of the namespace in which the qualifier type is to be created (Context Parameter)
NewQualifierType	QualifierType	Mandatory	Representation of the qualifier type to be created

3408

3409 **Operation output parameters:**

3410

Generic Name	Generic Type	Requirement	Description
QualifierTypePath	QualifierTypePath	Mandatory	Qualifier type path of the new qualifier type

3411

3412 **Description:**

3413 The *CreateQualifierType* operation creates a qualifier type in the namespace referenced by
 3414 *NamespacePath*, using the qualifier type representation specified in *NewQualifierType*, and returns
 3415 the qualifier type path of the new qualifier type.

3416 As defined in [DSP0004](#), any namespace needs to contain qualifier types for the meta qualifiers and
 3417 standard qualifiers, and may contain qualifier types for the optional qualifiers. Thus, creating these
 3418 qualifier types in a namespace inconsistently with their [DSP0004](#) definition will render that
 3419 namespace non-compliant to [DSP0004](#).

3420 **Preconditions:**

- 3421 • The namespace referenced by *NamespacePath* shall exist. If it does not exist, the operation
 3422 shall fail, indicating WIPG0204.
- 3423 • The qualifier type to be created shall not exist in the namespace referenced by
 3424 *NamespacePath*. If this is not satisfied, the operation shall fail, indicating WIPG0208.
- 3425 • If the qualifier type defined in *NewQualifierType* is one of the qualifiers defined in [DSP0004](#),
 3426 (i.e., meta, standard, and optional qualifiers), the definition of the qualifier in *NewQualifierType*
 3427 shall be consistent with the definition of the qualifier in [DSP0004](#). If this is not satisfied, the
 3428 operation shall fail, indicating WIPG0245.

3429 **Postconditions:**

- 3430 • The qualifier type shall have been created as defined in the Description paragraph for this
 3431 operation.
- 3432 • Requirements on ACID properties:
 - 3433 – Atomicity: Required
 - 3434 – Update Consistency: Required
 - 3435 – Isolation: Required
 - 3436 – Durability: Required

3437 **Error Messages:**
3438

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM server is shutting down	Optional	Infrastructure	
WIPG0240	WBEM server limits are exceeded	Optional	Infrastructure	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM server infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure	
WIPG0245	Qualifier type inconsistent with DSP0004	Mandatory	Infrastructure	
WIPG0243	Timeout	Optional	Infrastructure	
WIPG0227	Other failure	Optional	Infrastructure	

3439

3440 **6.9.5 EnumerateQualifierTypes**

3441 **Purpose:**

3442 Enumerate the qualifier types in a namespace.

3443 **Operation input parameters:**

3444

Generic Name	Generic Type	Requirement	Description
NamespacePath	NamespacePath	Mandatory	Namespace path of the namespace in which the qualifier types are to be enumerated (Context Parameter)

3445

3446 **Operation output parameters:**

3447

Generic Name	Generic Type	Requirement	Description
QualifierTypeList	QualifierTypeWithPath []	Mandatory	Sequence of the returned qualifier type representations and qualifier type paths

3448

3449 **Description:**

3450 The *EnumerateQualifierTypes* operation enumerates all qualifier types in the namespace referenced
 3451 by *NamespacePath*, and returns their representations and qualifier type paths.

3452 **Preconditions:**

- 3453 • The namespace referenced by *NamespacePath* shall exist. If it does not exist, the operation
 3454 shall fail, indicating WIPG0204.

3455 **Postconditions:**

- 3456 • The qualifier type representations and qualifier type paths shall have been returned as defined
 3457 in the Description paragraph for this operation.
- 3458 • Requirements on ACID properties:
- 3459 – Atomicity: N/A
 - 3460 – Update Consistency: N/A
 - 3461 – Isolation: Required at the level of single qualifier types, as defined in 5.8.
 - 3462 – Durability: N/A

3463 **Error Messages:**

3464

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM server is shutting down	Optional	Infrastructure	
WIPG0240	WBEM server limits are exceeded	Optional	Infrastructure	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM server infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure	
WIPG0243	Timeout	Optional	Infrastructure	
WIPG0227	Other failure	Optional	Infrastructure	

3465

3466 **6.10 Indication delivery operations**

3467 This subclause defines listener operations that deal with the delivery of indications.

3468 **6.10.1 DeliverIndication**

3469 **Purpose:**

3470 Deliver an indication to a listener.

3471 **Operation input parameters:**

3472

Generic Name	Generic Type	Requirement	Description
ListenerDestination	ListenerDestination	Mandatory	Address of the listener to which the indication will be delivered (see 5.4.20 for details) (Context Parameter)
Indication	InstanceSpecification	Mandatory	Representation of the indication instance

3473

3474 **Operation output parameters:**

3475 None.

3476 **Description:**

3477 The *DeliverIndication* listener operation delivers the indication specified by *Indication* to the listener
 3478 referenced by *ListenerDestination*.

3479 Reliable indication delivery as defined in DSP1054 is an optional part of the operation semantics.
 3480 Generic operations mappings shall state whether reliable indication delivery is supported.

3481 **Preconditions:**

- 3482 • None.

3483 **Postconditions:**

- 3484 • The indication shall have been delivered to the listener.
- 3485 • Requirements on ACID properties:
 - 3486 – Atomicity: N/A
 - 3487 – Update Consistency: N/A
 - 3488 – Isolation: N/A
 - 3489 – Durability: N/A

3490 **Error Messages:**

3491

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0250	WBEM listener is shutting down	Optional	Infrastructure	
WIPG0251	WBEM listener limits are exceeded	Optional	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure	
WIPG0243	Timeout	Optional	Infrastructure	This also covers timeout due to exhaustion of retries in reliable indication delivery (if supported).
WIPG0227	Other failure	Optional	Infrastructure	

3492

ANNEX A (informative)

Future operations

3493
3494
3495
3496

3497 This annex provides ideas for future operations or extensions to existing operations.

3498 **A.1 Test for property modifiability**

3499 Today, management profiles specify the modifiability of properties or an algorithm how to find out their
3500 modifiability at runtime. Usually, this includes the overhead of capability based mechanisms most of the
3501 time at the level of single properties. Because of this overhead, it is defined rarely in profiles and thus left
3502 to be decided by the implementation, with no defined way for a client to find out about it upfront.

3503 An operation (or an extension to an existing operation) that allows testing for modifiability of properties in
3504 a consistent way without depending on hard wired understanding of profile-defined modifiability or profile
3505 defined algorithms to find out modifiability would be a worthwhile extension.

3506 **A.2 Retrieval of associated instance graph**

3507 Today, a graph of associated instances can be retrieved only piece by piece, even distinguishing between
3508 retrieval of association instances (e.g., via *GetReferencingInstance...*) and associated instances (e.g., via
3509 *GetAssociatedInstance...*). Also, retrieving the associated instances associated by different associations
3510 may involve the invocation of multiple class implementations in typical CIMOM/provider based
3511 implementations, which could be optimized by having a single implementation of a more complex
3512 operation like the one proposed here.

3513 An operation would be helpful that can retrieve the graph of associated instances including their
3514 associations. Ideally, the operation would be able to traverse multiple association hops in one invocation.

3515 One possible definition of such operations could be:

3516 Direct retrieval: The *GetAssociatedGraphInstancesWithPath* operation traverses an association from an
3517 instance on a source end to instances on all of its far ends and returns the associated instances and their
3518 association instances, each together with their instance paths. This operation can be used to return one
3519 set of instances that would have otherwise required at least twice as many operations (one set to get the
3520 associations and another to get the related instances).

3521 Pulled retrieval: The *OpenAssociatedGraphInstancesWithPath* operation establishes and opens an
3522 enumeration session for enumerating instances that are associated with the specified source instance,
3523 and their association instances, including their instance paths. This operation can be used to return one
3524 set of instances that would have otherwise required at least twice as many operations (one set to get the
3525 associations and another to get the related instances).

ANNEX B (informative)

Changed generic operation names

3526
3527
3528
3529

3530 Versions 1.0.2 (and 1.1.0) of this document changed the names of the generic operations in order to align
3531 them with the operation names of the CIM-XML protocol (see [DSP0200](#)) that are used in current
3532 management profiles. This change allows management profiles to more easily use the generic operation
3533 names, which is required when using the condensed format defined in [DSP1001](#) V1.1 or when migrating
3534 profiles to become machine readable (see [DSP2023](#)).

3535 Note that the new generic operations are not always 1:1 with the CIM-XML operations: For example, in
3536 CIM-XML, the association operations are one set of operations covering both instance and class level,
3537 while in generic operations, class and instance level operations continue to be separated.

3538 This name change is incompatible for management profiles that specified operation requirements using
3539 the old generic operation names. There is only one such DMTF profile ([DSP1054 Version 1.2](#)). However,
3540 it is not an incompatible change for implementations of such profiles because the names of generic
3541 operations are not visible in the implementation; they remain at the specification level. All APIs and
3542 protocols the DMTF knows about do not currently use the generic operation names in their specifications
3543 or in their implementations (except for mappings between the APIs or protocols and generic operations).

3544 Table B-1 lists the old and new operation names for all operations defined in this document.

3545

Table B-1 – Changed generic operation names

New operation name (starting with V1.0.2 and V1.1.0)	Old operation name (in V1.0.0 and V1.0.1)	Name Changed	Description
GetInstance	GetInstance	no	See 6.3.1
DeleteInstance	DeleteInstance	no	See 6.3.2
ModifyInstance	ModifyInstance	no	See 6.3.3
CreateInstance	CreateInstance	no	See 6.3.4
EnumerateInstances (deprecated)	GetClassInstancesWithPath	yes	See 6.4.1
EnumerateInstanceNames (deprecated)	GetClassInstancePaths	yes	See 6.4.2
Associators (deprecated)	GetAssociatedInstancesWithPath	yes	See 6.4.3
AssociatorNames (deprecated)	GetAssociatedInstancePaths	yes	See 6.4.4
References (deprecated)	GetReferencingInstancesWithPath	yes	See 6.4.5
ReferenceNames (deprecated)	GetReferencingInstancePaths	yes	See 6.4.6
OpenEnumerateInstances	OpenClassInstancesWithPath	yes	See 6.5.3
OpenEnumerateInstancePaths (deprecated)	OpenClassInstancePaths	yes	See 6.5.4
OpenAssociators	OpenAssociatedInstancesWithPath	yes	See 6.5.5
OpenAssociatorPaths (deprecated)	OpenAssociatedInstancePaths	yes	See 6.5.6
OpenReferences	OpenReferencingInstancesWithPath	yes	See 6.5.7
OpenReferencePaths (deprecated)	OpenReferencingInstancePaths	yes	See 6.5.8
OpenQueryInstances	OpenQueryInstances	no	See 6.5.9

New operation name (starting with V1.0.2 and V1.1.0)	Old operation name (in V1.0.0 and V1.0.1)	Name Changed	Description
PullInstancesWithPath	PullInstancesWithPath	no	See 6.5.11
PullInstancePaths (deprecated)	PullInstancePaths	no	See 6.5.11
PullInstances	PullInstances	no	See 6.5.13
CloseEnumeration	CloseEnumeration	no	See 6.5.14
EnumerationCount (deprecated)	EnumerationCount	no	See 6.5.15
InvokeMethod	InvokeMethod	no	See 6.6.1
InvokeStaticMethod	InvokeStaticMethod	no	See 6.6.2
GetClass	GetClass	no	See 6.7.1
DeleteClass	DeleteClass	no	See 6.7.2
ModifyClass	ModifyClass	no	See 6.7.3
CreateClass	CreateClass	no	See 6.7.4
EnumerateClasses (1)	GetTopClassesWithPath	yes	See 6.8.1 (1)
EnumerateClassNames (2)	GetTopClassPaths	yes	See 6.8.2 (2)
EnumerateClasses (1)	GetSubClassesWithPath	yes	See 6.8.1 (1)
EnumerateClassNames (2)	GetSubClassPaths	yes	See 6.8.2 (2)
AssociatorClasses	GetAssociatedClassesWithPath	yes	See 6.8.3
AssociatorClassPaths	GetAssociatedClassPaths	yes	See 6.8.4
ReferenceClasses	GetReferencingClassesWithPath	yes	See 6.8.5
ReferenceClassPaths	GetReferencingClassPaths	yes	See 6.8.6
GetQualifierType	GetQualifierType	no	See 6.9.1
DeleteQualifierType	DeleteQualifierType	no	See 6.9.2
ModifyQualifierType	ModifyQualifierType	no	See 6.9.3
CreateQualifierType	CreateQualifierType	no	See 6.9.4
EnumerateQualifierTypes	EnumerateQualifierTypesWithPath	yes	See 6.9.5

3546

3547 Notes:

- 3548 (1) The old *GetTopClassesWithPath* and *GetSubClassesWithPath* operations have been merged
3549 into the new *EnumerateClasses* operation that covers both top classes and subclasses.
- 3550 (2) The old *GetTopClassPaths* and *GetSubClassPaths* operations have been merged into the new
3551 *EnumerateClassNames* operation that covers both top classes and subclasses.

ANNEX C (normative)

Cross-namespace associations

3556 This annex describes cross-namespace associations, in order to define which instances and classes exist
 3557 in which namespace in such a scenario, and what is to be returned by association traversal operations.
 3558 This annex reflects the preconditions stated for the association traversal operations in this specification,
 3559 but it defines additional rules for conforming implementations and is therefore normative.

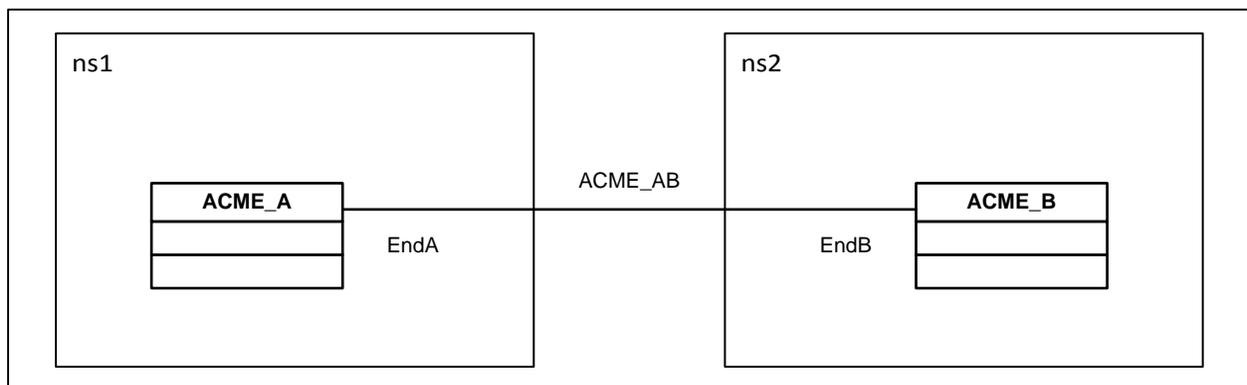
3560 In this annex, classes in a particular namespace are referred to using the syntax `<ns-name>::<class-`
 3561 `name>` where `<ns-name>` is the namespace name and `<class-name>` is the class name. Instances in a
 3562 particular namespace are referred to using the syntax `<ns-name>::<inst-name>` where `<ns-name>`
 3563 is again the namespace name and `<inst-name>` is the name of the instance as stated in the diagram
 3564 (which is purely a diagramming name and has nothing to do with its keys).

3565 In this version of this document, this annex only covers the simple case of a binary association where
 3566 both sides use the same schema version. More complex cases, e.g. of associations with more than two
 3567 ends, or with schema different versions, are possible, but not covered in this version.

3568 C.1 Binary association using same schema version

3569 This subclause discusses a binary association (that is, an association with two ends) that crosses
 3570 namespaces, and the two namespaces contain the same version of the schema.

3571 Figure C-2 is a UML class diagram showing the classes used by this scenario, in the typical drawing
 3572 notation used in management profiles:



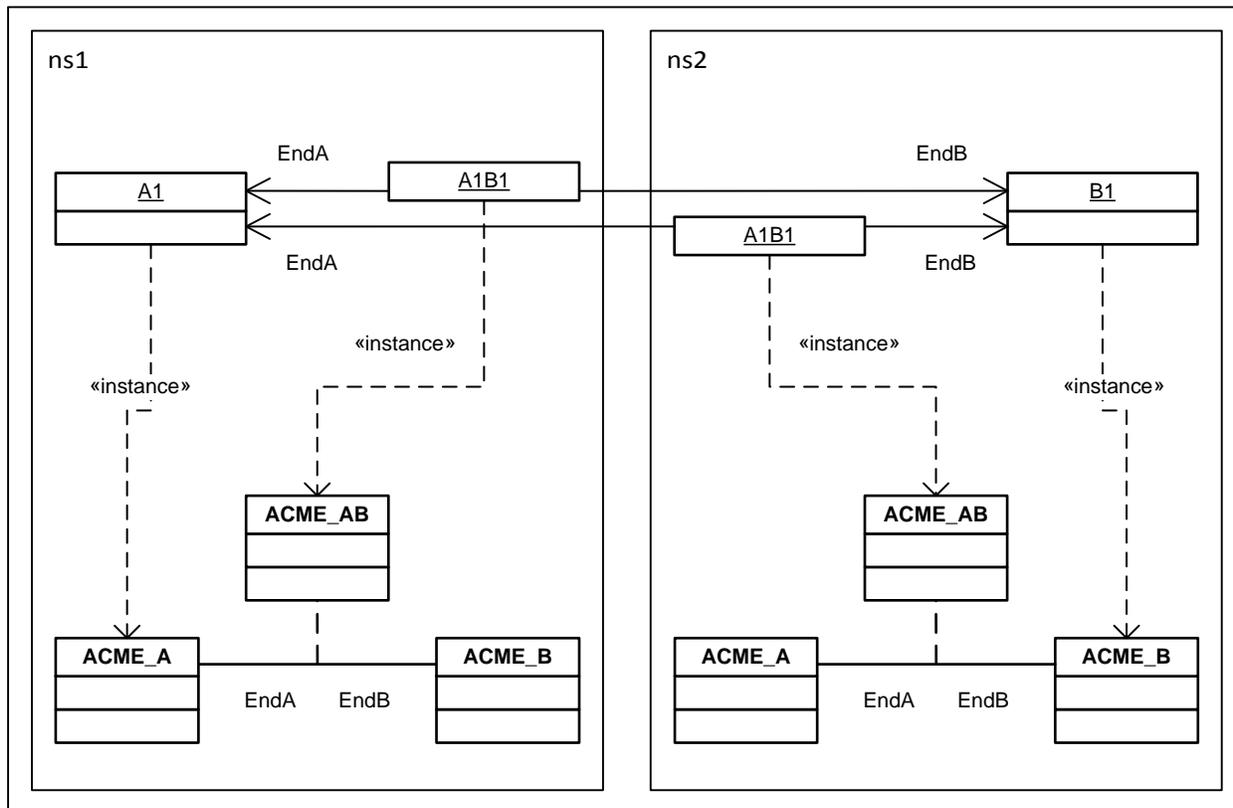
3573
 3574

3575 Figure C-2 – Typical profile representation of binary association crossing namespaces

3576 The namespaces (in this scenario, `ns1` and `ns2`) are shown as boxes around a number of classes used
 3577 in the management profile. Class `ACME_A` is in namespace `ns1`, class `ACME_B` in namespace `ns2`.
 3578 Association `ACME_AB` crosses between these two namespaces and is therefore termed a *cross-*
 3579 *namespace association*.

3580 This way of drawing the situation leaves it open exactly which classes and instances exist in each of the
 3581 namespaces, and which of them are returned by instance-level and class-level association traversal
 3582 operations.

3583 Figure C-3 is a UML structure diagram showing the classes and instances in a WBEM server that need to
 3584 exist when the classes shown in Figure C-2 are implemented for bidirectional association traversal.



3585
 3586

3587 **Figure C-3 – Binary association: WBEM server objects for bidirectional traversal**

3588 The upper part of the figure shows instances, the lower part shows classes. Both of these are objects in a
 3589 particular namespace of a WBEM server. Note that every object in this diagram is contained in a
 3590 namespace. This is consistent with [DSP0004](#) which defines that the name of every object (class,
 3591 instance, qualifier type) has a namespace path component. As a result, the instance A1B1 of the cross-
 3592 namespace association ACME_AB appears in each of the two namespaces; in a way, it is duplicated.

3593 **Rule:** Conformant implementations of bidirectional association traversal across namespaces shall have
 3594 any such bidirectional cross-namespace association instances exist in both namespaces, and shall have
 3595 the instances associated through such cross-namespace associations exist in only one namespace.

3596 Enumerating the instances of the association class ACME_AB in namespace ns1 (e.g., with the
 3597 EnumerateInstances operation) returns instance ns1::A1B1, and enumerating the instances of
 3598 ACME_AB in namespace ns2 returns instance ns2::A1B1. This means that the association instances act
 3599 like any other instances: They can be enumerated (if the operation is implemented) and that enumeration
 3600 is scoped to a particular namespace. The instances ns1::A1B1 and ns2::A1B1 are distinct instances,
 3601 because their namespace path is different.

3602 The instances of the associated classes ACME_A1 and ACME_B1 appear only in their respective
 3603 namespaces; they are not duplicated. As a result, enumerating the instances of ACME_A in namespace
 3604 ns1 returns ns1::A1, and enumerating the instances of ACME_A in ns2 returns no instances (the
 3605 EnumerateInstances operation still succeeds, because the class ACME_A exists in ns2).

3606 The association traversal operations work in both directions in this scenario:

3607 Traversing association `ACME_AB` starting from instance `ns1::A1` using the `Associators` operation results
3608 in instance `ns2::B1`, and traversing association `ACME_AB` starting from instance `ns2::B1` using the
3609 `Associators` operation results in instance `ns1::A1`. Because this behavior can be determined from the
3610 description of these operations for the single-namespace case, no special rule for the cross-namespace
3611 case has been defined.

3612 Because of the duplication of association instances in both namespaces, the situation is not intuitively
3613 clear for the `References` operation and other association-returning operations: The association instances
3614 `ns1::A1B1` and `ns2::A1B1` both reference the instance `ns1::A1`, so from a perspective of following
3615 the specified behavior for this operation by the letter, one can argue that both instances need to be
3616 returned, because they both exist and both reference the source instance. However, because these two
3617 instances are logically the same, a client would need to reduce the result set by eliminating such logical
3618 duplicates. Therefore, this annex defines the following restricting rule:

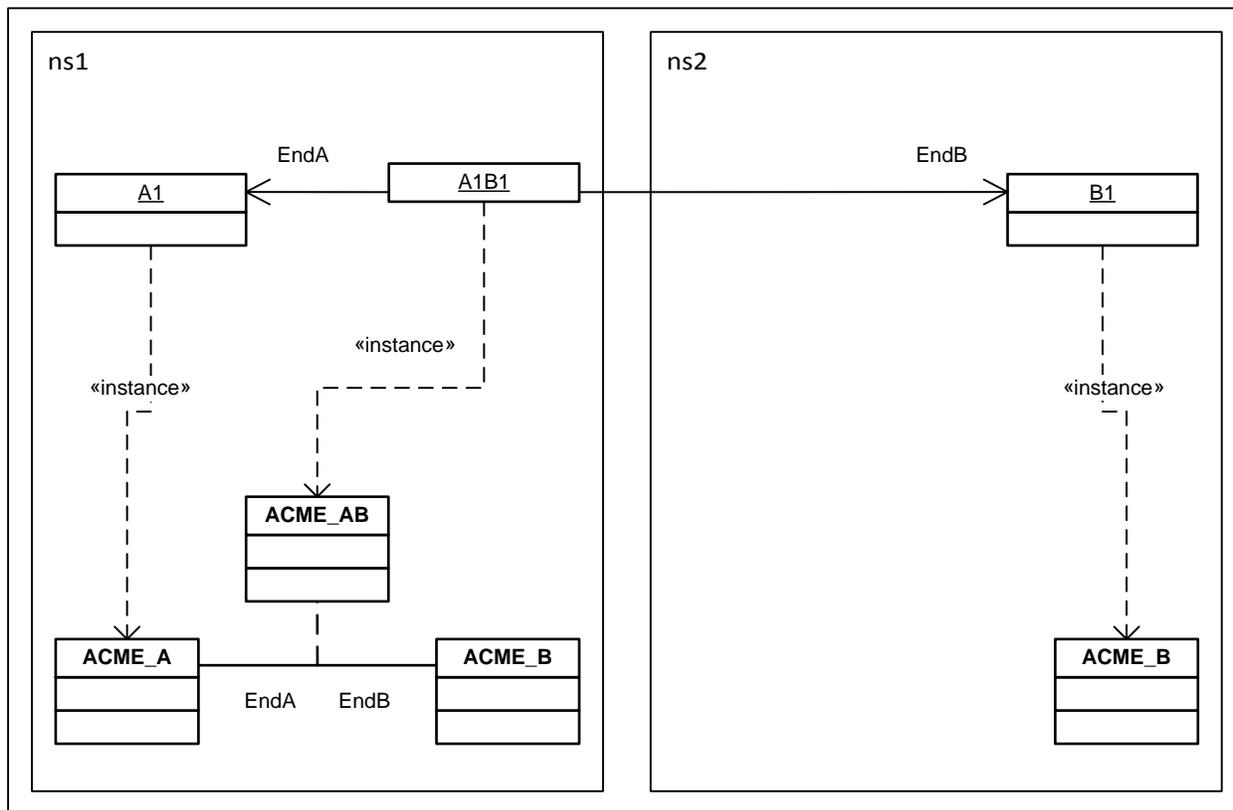
3619 **Rule:** Conformant implementations of the `References` operation and other association-returning
3620 operations (such as `ReferenceNames` and `OpenReferences`) shall return only association instances that
3621 exist in the namespace of the source instance (even if a duplicated association instance exists in the
3622 other namespace).

3623 For classes, the existence requirements are driven by their role as a creation class, and their role as a
3624 declared target of a reference in an association class. In Figure C-3, the four classes that are the target of
3625 the `<<instance>>` dependency need to exist because they have instances in their namespace.
3626 Because the references in an association class only declare their targeted class but not their targeted
3627 namespace, schema consistency rules require that all classes referenced by an association class exist as
3628 objects in the same namespace as the association class. As a result, class `ACME_A` in addition needs to
3629 exist in `ns2`, and class `ACME_B` in addition needs to exist in `ns1`. Note that this is driven by consistency
3630 rules within a schema in a namespace, and is independent of whether or not class-level association
3631 traversal operations are supported. As a result, no additional rule needs to be defined for the existence of
3632 class objects in a cross-namespace case.

3633 Because of the limitation that class-level references do not declare a target namespace, this annex
3634 defines the following rule for the behavior of the class-level operations:

3635 **Rule:** Conformant implementations of the `AssociatorClasses` and `ReferenceClasses` operation shall
3636 return only classes that exist in the namespace of the source class; they never cross namespace
3637 boundaries.

3638 Figure C-4 shows the classes and instances in a WBEM server that need to exist when the classes
3639 shown in Figure C-2 are implemented for unidirectional association traversal in the direction from `ns1` to
3640 `ns2`:



3641
3642

3643 **Figure C-4 – Binary association: WBEM server objects for unidirectional traversal**

3644 In this case, the association instance `A1B1` only exists in namespace `ns1`, where traversal starts from.

3645 **Rule:** Conformant implementations of unidirectional association traversal across namespaces shall have
 3646 any such unidirectional cross-namespace association instances exist in only the source namespace
 3647 where traversal starts from, and shall have the instances associated through such cross-namespace
 3648 associations exist in only one namespace.

3649 Because there is no instance `A1B1` in namespace `ns2`, there is no need for the classes `ACME_AB` and
 3650 `ACME_A` to exist in `ns2`. As a result, namespace `ns2`, is "logically unaware" that namespace `ns1` can
 3651 traverse into it. Whether this implies "implementation unawareness" depends on the type of WBEM server
 3652 infrastructure that is used.

ANNEX D (informative)

Change log

3653
3654
3655
3656

3657

Version	Date	Description
1.0.0	2010-04-22	
1.1.0	2015-02-19	<p>Published as DMTF Standard with the following changes:</p> <ul style="list-style-type: none"> • Fixed an error in the description of the IncludeInheritedElements parameter of the GetSubClassesWithPath operation (it is based on the specified class, not on the returned classes). • Clarified why the GetTopClassesWithPath operation does not have an IncludeInheritedElements parameter. • Deprecated the IncludeProperties parameter of the GetClass, GetAssociatedClassesWithPath, and GetReferencingClassesWithPath operations, with no replacement. • In CreateInstance, fixed that property default values are now treated as an initialization constraint, together with the PropertyConstraint qualifier and constraints defined in management profiles. • In GetClass, fixed the Class output parameter to now be of type ClassSpecification (it was ClassSpecificationWithPath, returning the input class path again). • Added support for indications. • Added IncludeInheritedElements parameter to getClass. • Clarified that class origin information indicates the leaf-most class defining the element, in override situations, consistent with DSP0200 1.4. • Narrowed the definition of "WBEM protocol" in that it needs to conform to generic operations. • Improved the description of the interaction model and distinction between generic operations and WBEM protocol. • Updated the minor versions of several normative references, added DSP1054 as a normative reference, and moved DSP0228 to the Bibliography. • Wording improvements throughout the document. • From 1.0.2: Errata: Changed the names of the generic operations to be aligned with the CIM-XML operation names. See ANNEX B for details. • From 1.0.2: For the PullInstances operation, fixed an incorrect occurrence of its name, and an error in its description where it was incorrectly stated that it would return instances with path. • Added references to CIM-RS specifications. • Added requirement for WBEM protocols to support FQL (Filter Query Language). • Deprecated non-pulled instance operations that have pulled equivalents (EnumerateInstances, EnumerateInstanceNames, Associators, AssociatorNames, References, ReferenceNames). • Deprecated pulled instance operations returning instance paths (OpenEnumerateInstancePaths, OpenAssociatorPaths, OpenReferencePaths). • Deprecated EnumerationCount operation. • Deprecated IncludeClassOrigin input parameter on any instance operations (GetInstance, EnumerateInstances, Associators, References, OpenEnumerateInstances, OpenAssociators, OpenReferences). • Clarified which components have to be present in any object paths (namespace, instance, class, qualifier type). • Clarified requirements for existence of classes and namespaces of any class-level and instance-level (pulled and non-pulled) association operations, by

Version	Date	Description
		<p>adding according preconditions.</p> <ul style="list-style-type: none"> • Errata: Fixed the missing WIPG0214 (Class not found) in pulled association operations. • Errata: Changed the behavior of (pulled and non-pulled) association operations in case the source instance does not exist, from failing to succeeding with an empty result set, in order to be aligned with the behavior of the corresponding CIM-XML operations. As a result, removed WIPG0213 (Instance not found) from their set of allowable error messages. • Errata: Removed the AssociatedClassName and AssociatedRoleName filters from (pulled and non-pulled) reference operations, in order to be aligned with the filtering abilities of the corresponding CIM-XML operations. • Errata: Changed the name of parameter RoleName to SourceRoleName, of the class-level association operations, for consistency with the corresponding instance-level operations. • Added WIPG0240 (WBEM server limits are exceeded) to all operations that did not have it yet. • Added WIPG0214 (Class not found) to InvokeMethod operation. • Added ANNEX C, defining normative rules for cross-namespace associations. • Terms and abbreviations are now based on DSP0198.

3658

Bibliography

- 3659
- 3660 DMTF DSP0200, *CIM Operations over HTTP 1.3*,
3661 http://www.dmtf.org/standards/published_documents/DSP0200_1.3.pdf
- 3662 DMTF DSP0201, *Representation of CIM in XML 2.3*,
3663 http://www.dmtf.org/standards/published_documents/DSP0201_2.3.pdf
- 3664 DMTF DSP0202, *CIM Query Language Specification 1.0*,
3665 http://www.dmtf.org/standards/published_documents/DSP0202_1.0.pdf
- 3666 DMTF DSP0203, *DTD for Representation of CIM in XML 2.3*,
3667 http://www.dmtf.org/standards/published_documents/DSP0203_2.3.dtd
- 3668 DMTF DSP0210, *CIM-RS Protocol 2.0*,
3669 http://www.dmtf.org/standards/published_documents/DSP0210_2.0.pdf
- 3670 DMTF DSP0211, *CIM-RS Payload Representation in JSON 2.0*,
3671 http://www.dmtf.org/standards/published_documents/DSP0211_2.0.pdf
- 3672 DMTF DSP0214, *Server Management Command Line Protocol Specification 1.0*,
3673 http://www.dmtf.org/standards/published_documents/DSP0214_1.0.pdf
- 3674 DMTF DSP0226, *Web Services for Management 1.0*,
3675 http://www.dmtf.org/standards/published_documents/DSP0226_1.0.pdf
- 3676 DMTF DSP0227, *WS-Management CIM Binding Specification 1.0*,
3677 http://www.dmtf.org/standards/published_documents/DSP0227_1.0.pdf
- 3678 DMTF DSP0228, *Message Registry XML Schema 1.1*,
3679 http://schemas.dmtf.org/wbem/messageregistry/1/dsp0228_1.1.xsd
- 3680 DMTF DSP0230, *WS-CIM Mapping Specification 1.0*,
3681 http://www.dmtf.org/standards/published_documents/DSP0230_1.0.pdf
- 3682 DMTF DSP1001, *Management Profile Specification Usage Guide 1.2*,
3683 http://www.dmtf.org/standards/published_documents/DSP1001_1.2.pdf
- 3684 DMTF DSP8028, *Management Profile XML Schema 1.1*,
3685 http://schemas.dmtf.org/wbem/mgmtprofile/1/dsp8028_1.1.xsd
- 3686 JCP JSR-48, *Java Community Process JSR-48: WBEM servers Specification*, not yet published,
3687 <http://jcp.org/en/jsr/detail?id=48>
- 3688 The Open Group CMPI, *Systems Management: Common Manageability Programming Interface (CMPI),*
3689 *Issue 2.0*, <http://www.opengroup.org/bookstore/catalog/c061.htm>