



1

2

3

4

Document number: DSP0223

Date: 2013-10-22

Version: 1.0.2

5 **Generic Operations**

6 **Document type: Specification**

7 **Document status: DMTF Standard**

8 **Document language: en-US**

9

10 Copyright notice

11 Copyright © 2007–2013 Distributed Management Task Force, Inc. (DMTF). All rights reserved.

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

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26 withdrawn or modified after publication, and shall be indemnified and held harmless by any party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

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

32

Contents

33	Foreword	6
34	Acknowledgements	6
35	Document conventions	6
36	Typographical conventions	6
37	Experimental material	6
38	1 Scope	7
39	2 Normative references	7
40	3 Terms and definitions	8
41	4 Symbols and abbreviated terms	11
42	5 Concepts	11
43	5.1 Generic operations model	11
44	5.2 Generic operations mappings	12
45	5.2.1 Overview	12
46	5.2.2 Recommendations	13
47	5.3 Conformance to generic operations	13
48	5.3.1 Conformance of WBEM protocols or APIs	14
49	5.3.2 Conformance of WBEM operations or API calls	14
50	5.3.3 Requirement levels for operation parameters	14
51	5.4 Generic types	15
52	5.4.1 CIM data types	15
53	5.4.2 NamespacePath	15
54	5.4.3 InstancePath	15
55	5.4.4 ClassPath	15
56	5.4.5 QualifierTypePath	15
57	5.4.6 InstanceSpecification	16
58	5.4.7 ClassSpecification	16
59	5.4.8 QualifierType	17
60	5.4.9 InstanceSpecificationWithPath	17
61	5.4.10 ClassSpecificationWithPath	17
62	5.4.11 QualifierTypeWithPath	18
63	5.4.12 ClassName	18
64	5.4.13 PropertyName	18
65	5.4.14 MethodName	18
66	5.4.15 ParameterValue	18
67	5.4.16 ReturnValue	18
68	5.4.17 QueryString	18
69	5.4.18 QueryLanguage	18
70	5.4.19 EnumerationContext	18
71	5.5 Success and failure	19
72	5.6 Preconditions and postconditions	19
73	5.7 Generic error messages	19
74	5.8 Consistency model	20
75	5.8.1 Definition of ACID properties	20
76	5.8.2 Time consistency within a CIM instance	21
77	5.8.3 Staleness of information returned	21
78	5.8.4 Isolation between operations	21
79	5.8.5 Duplicate return of CIM objects or object paths	22
80	5.8.6 Time consistency between returned CIM objects	22
81	5.8.7 Order of returned CIM objects	22
82	5.8.8 Validity of returned object paths	22
83	5.8.9 Effects of deleting an instance	23
84	6 Generic operations	25

85	6.1	Description format.....	26
86	6.2	Common operation parameters for all operations	28
87	6.2.1	IncludeClassOrigin.....	28
88	6.2.2	IncludeQualifiers	28
89	6.2.3	<element>List	28
90	6.3	Instance operations.....	29
91	6.3.1	GetInstance.....	29
92	6.3.2	DeleteInstance.....	31
93	6.3.3	ModifyInstance.....	33
94	6.3.4	CreateInstance.....	35
95	6.4	Direct instance enumeration operations	38
96	6.4.1	EnumerateInstances	38
97	6.4.2	EnumerateInstanceNames	40
98	6.4.3	Associators	42
99	6.4.4	AssociatorNames.....	45
100	6.4.5	References.....	48
101	6.4.6	ReferenceNames	51
102	6.5	Pulled instance enumeration operations.....	54
103	6.5.1	General behavioral rules.....	54
104	6.5.2	Common operation parameters for the open operations	56
105	6.5.3	OpenEnumerateInstances	58
106	6.5.4	OpenEnumerateInstancePaths.....	62
107	6.5.5	OpenAssociators.....	65
108	6.5.6	OpenAssociatorPaths	69
109	6.5.7	OpenReferences.....	73
110	6.5.8	OpenReferencePaths	78
111	6.5.9	OpenQueryInstances	81
112	6.5.10	Common operation parameters for the pull operations	84
113	6.5.11	PullInstancesWithPath	85
114	6.5.12	PullInstancePaths	87
115	6.5.13	PullInstances.....	89
116	6.5.14	CloseEnumeration	91
117	6.5.15	EnumerationCount.....	92
118	6.6	Method invocation operations.....	94
119	6.6.1	InvokeMethod	94
120	6.6.2	InvokeStaticMethod	96
121	6.7	Class operations	98
122	6.7.1	GetClass	98
123	6.7.2	DeleteClass.....	99
124	6.7.3	ModifyClass	103
125	6.7.4	CreateClass	105
126	6.8	Class enumeration operations	106
127	6.8.1	EnumerateClasses.....	107
128	6.8.2	EnumerateClassNames.....	109
129	6.8.3	AssociatorClasses	111
130	6.8.4	AssociatorClassPaths	113
131	6.8.5	ReferenceClasses.....	115
132	6.8.6	ReferenceClassPaths	118
133	6.9	Qualifier type operations	120
134	6.9.1	GetQualifierType.....	120
135	6.9.2	DeleteQualifierType	121
136	6.9.3	ModifyQualifierType	123
137	6.9.4	CreateQualifierType.....	124
138	6.9.5	EnumerateQualifierTypes	126
139	ANNEX A (informative)	Future operations	128
140	A.1	Test for property modifiability.....	128

141	A.2 Retrieval of associated instance graph.....	128
142	ANNEX B (informative) Changed generic operation names.....	129
143	ANNEX C (informative) Change log.....	131
144	Bibliography	132
145		
146	Figures	
147	Figure 1 – Generic operations model.....	12
148	Figure 2 – Generic operations mappings	13
149		
150	Tables	
151	Table 1 – List of generic operations.....	25
152	Table 2 – Changed generic operation names.....	129
153		

155

Foreword

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

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

160 Acknowledgements

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

- 162 • Andreas Maier, IBM (editor)
- 163 • Jim Davis, WBEM Solutions
- 164 • George Ericson, EMC
- 165 • Steve Hand, Symantec
- 166 • Jon Hass, Dell
- 167 • Lawrence Lamers, VMware

168 Document conventions

169 Typographical conventions

170 The following typographical conventions are used in this document:

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

176 Experimental material

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

182 The following typographical convention indicates experimental material:

183 **EXPERIMENTAL**

184 Experimental material appears here.

185 **EXPERIMENTAL**

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

188

189

Generic Operations

1 Scope

WBEM is a set of DMTF standards that define how CIM modeled resources can be discovered, accessed and manipulated. DMTF defines a number of WBEM protocols for this purpose:

- CIM-XML: The protocol defined in the CIM Operations over HTTP Specification [DSP0200](#), the Representation of CIM in XML Specification [DSP0201](#) and the DTD for Representation of CIM in XML [DSP0203](#).
- WS-MAN: The usage of the WS-Management protocol for CIM, as defined in the WS-Management CIM Binding Specification [DSP0227](#), the WS-CIM Mapping Specification [DSP0230](#), the Web Services for Management Specification [DSP0226](#), and other underlying Web Services specifications.
- SM-CLP: The protocol defined in the Server Management Command Line Protocol Specification [DSP0214](#), covering the core of the protocol common for all management profiles, and SM-CLP mapping specifications for each management profile, covering profile specific aspects of the protocol such as verbs for extrinsic methods.

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

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

- Future releases of CIM management profile specifications can define the support for intrinsic operations by referencing generic operations. Currently, they do that by referencing the operations defined for the CIM-XML protocol. Using generic operations allows the management profile specifications to become independent of protocols.
- Future and existing WBEM protocols can define their operations conformant to the generic operations. This drives more commonality across these protocols, and consequently makes it easier to support multiple protocols in client applications, server side instrumentation, and mapping bridges between protocols.
- Client APIs, server APIs and provider APIs can define their API calls conformant to the generic operations. This drives more commonality across these APIs and between these APIs and WBEM protocols, and consequently makes it easier to support multiple protocols with the same API in client libraries and server side instrumentation (e.g., provider APIs).

2 Normative references

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

DMTF DSP0004, *CIM Infrastructure Specification 2.6*,
http://www.dmtf.org/standards/published_documents/DSP0004_2.6.pdf

DMTF DSP0207, *WBEM URI Mapping 1.0*,
http://www.dmtf.org/standards/published_documents/DSP0207_1.0.pdf

230 DMTF DSP0228, *Message Registry XML Schema 1.1*,
231 http://schemas.dmtf.org/wbem/messageregistry/1/dsp0228_1.1.xsd

232 DMTF DSP8016, *WBEM Operations Message Registry 1.0*,
233 http://schemas.dmtf.org/wbem/messageregistry/1/dsp8016_1.0.xml

234 ISO/IEC Directives, Part 2:2004, *Rules for the structure and drafting of International Standards*,
235 <http://isotc.iso.org/livelink/livelink?func=ll&objId=4230456&objAction=browse>

236 **3 Terms and definitions**

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

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

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

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

250 The terms defined in [DSP0004](#) apply to this specification. The following additional terms are used in this
251 document.

252 **3.1**

253 **class path**

254 a special kind of object path addressing a CIM class that is accessible through a WBEM server
255 For details, see [DSP0004](#).

256 **3.2**

257 **creation class**

258 the creation class of a CIM instance is the most derived class the instance is of
259 For a complete definition, see [DSP0004](#).

260 **3.3**

261 **duplicate object**

262 objects in a result set that have duplicate object paths

263 **3.4**

264 **duplicate object path**

265 object paths in a result set that reference the same CIM object accessible through the WBEM server

266 **3.5**

267 **effective qualifier value**

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

- 271 **3.6**
272 **exposed elements of a class**
273 The set of schema elements exposed by a class (i.e., properties and methods) is the union of the set of
274 elements defined in the class and the set of inherited elements that are not overridden in the class. For a
275 complete definition, see [DSP0004](#).
- 276 **3.7**
277 **generic operation**
278 a generic operation as defined in this specification
- 279 **3.8**
280 **generic operations mapping**
281 a mapping of generic operations to the operations of some other protocol (e.g., WBEM operations) or to
282 the calls of some API, as defined in 5.2
- 283 **3.9**
284 **instance path**
285 a special kind of object path addressing a CIM instance that is accessible through a WBEM server
286 For details, see [DSP0004](#).
- 287 **3.10**
288 **isolation**
289 the set of behaviors that describe how the execution of an operation affects the execution of another,
290 concurrent operation, as defined in 5.8.4
- 291 **3.11**
292 **management profile**
293 a management profile as defined in [DSP1001](#)
294 As used in this specification, the term includes all possible owners of such profiles, including other
295 standards organizations and vendors.
- 296 **3.12**
297 **namespace path**
298 a special kind of object path addressing a CIM namespace that is accessible through a WBEM server
299 For details, see [DSP0004](#).
- 300 **3.13**
301 **object**
302 a class, instance, qualifier type or namespace that is accessible through a WBEM server
303 For details, see [DSP0004](#).
- 304 **3.14**
305 **object path**
306 the address of an object that is accessible through a WBEM server
307 For details, see [DSP0004](#).
- 308 **3.15**
309 **qualifier type path**
310 a special kind of object path addressing a CIM qualifier type that is accessible through a WBEM server
311 For details, see [DSP0004](#).

- 312 **3.16**
313 **volatile property**
314 a property in a CIM 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 **3.17**
317 **WBEM client**
318 a CIM client (see [DSP0004](#)) that supports a WBEM protocol
319 A WBEM client originates WBEM operations for processing by a WBEM server. This definition does not
320 imply any particular implementation architecture or scope, such as a client library component or an entire
321 management application. For details, see 5.1.
- 322 **3.18**
323 **WBEM indication**
324 an interaction within a WBEM protocol that is originated on a WBEM server and processed by a WBEM
325 listener
326 This release of this specification does not cover WBEM indications.
- 327 **3.19**
328 **WBEM listener**
329 a CIM listener (see [DSP0004](#)) that supports a WBEM protocol
330 A WBEM listener processes WBEM indications originated by a WBEM server. This definition does not
331 imply any particular implementation architecture or scope, such as a standalone demon component or an
332 entire management application.
333 This release of this specification does not cover WBEM listeners.
- 334 **3.20**
335 **WBEM operation**
336 an interaction within a WBEM protocol that is originated by a WBEM client and processed by a WBEM
337 server
338 For details, see 5.1.
- 339 **3.21**
340 **WBEM protocol**
341 a communications protocol between WBEM client, WBEM server and WBEM listener
342 A WBEM protocol defines how the WBEM operations and WBEM indications work, on top of an
343 underlying protocol layer (for example, HTTP, SOAP, or TCP). For details, see 5.1.
- 344 **3.22**
345 **WBEM protocol mapping**
346 a mapping of generic operations to a WBEM protocol, as defined in 5.2
- 347 **3.23**
348 **WBEM server**
349 a CIM server (see [DSP0004](#)) that supports a WBEM protocol
350 A WBEM server processes WBEM operations originated by a WBEM client, and originates WBEM
351 indications for processing by a WBEM listener. This definition does not imply any particular
352 implementation architecture, such as a separation into a CIMOM and provider components. For details,
353 see 5.1.

354 **4 Symbols and abbreviated terms**

355 The symbols and abbreviations defined in [DSP0004](#) apply to this specification. The following additional
356 abbreviations are used in this document.

357 **4.1**

358 **API**

359 Application Programming Interface

360 **4.2**

361 **CIM**

362 Common Information Model, defined by DMTF

363 **4.3**

364 **CQL**

365 CIM Query Language, defined in [DSP0202](#)

366 **4.4**

367 **HTTP**

368 Hyper Text Transfer Protocol, defined by W3C

369 **4.5**

370 **UML**

371 Unified Modeling Language, defined by OMG

372 **4.6**

373 **WBEM**

374 Web Based Enterprise Management, defined by DMTF

375 **4.7**

376 **XML**

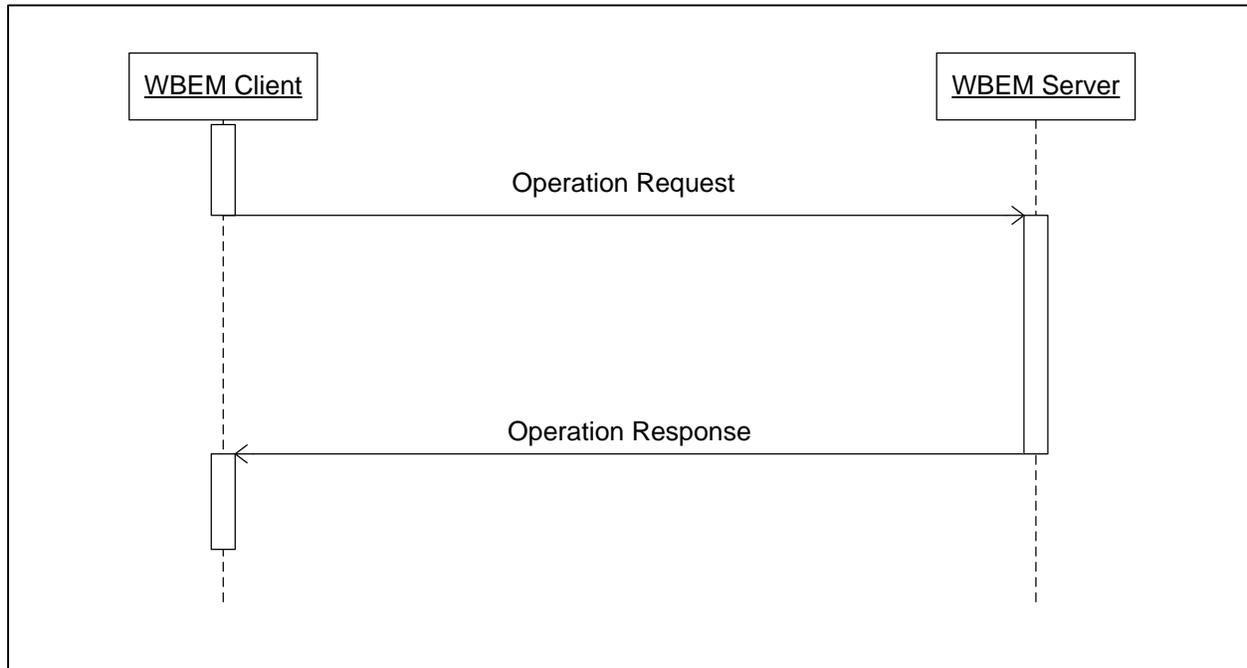
377 Extensible Markup Language, defined by W3C

378 **5 Concepts**

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

380 **5.1 Generic operations model**

381 Figure 1 shows the generic operations model using a UML sequence diagram:



382
383

384

Figure 1 – Generic operations model

385 In the generic operations model, *operations* are logical actions directed from a WBEM client to a WBEM
386 server. An *operation request* is sent from the client to the service when invoking the operation and an
387 *operation response* is sent back from the service to the client upon completion of the operation.

388 At the level of generic operations, any *input parameters* are part of the operation request, and any *output*
389 *parameters* are part of the operation response. A WBEM protocol may choose to do that differently, for
390 example by pushing some of the input parameters to the service side in the form of options that are set,
391 and that are used during the processing of subsequent operations.

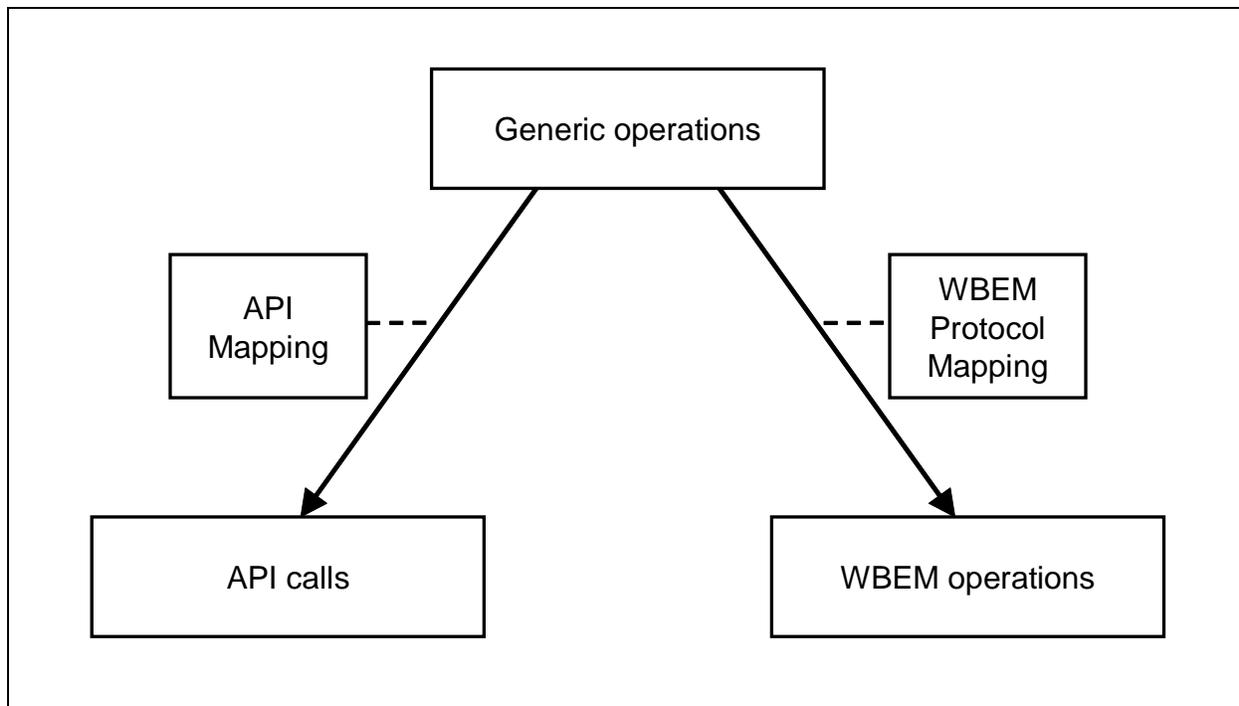
392 The operation request and operation response at the level of generic operations do not necessarily need
393 to correspond directly to messages that are flowing at the level of the WBEM protocol. For example, the
394 operation response may be delivered asynchronously at the level of the WBEM protocol.

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

399 5.2 Generic operations mappings

400 5.2.1 Overview

401 Figure 2 shows mappings of generic operations to WBEM protocols and APIs. These mappings allow
402 determining which WBEM operations or API calls need to be implemented for a particular generic
403 operation to be supported. This is used for example when implementing management profiles that specify
404 provisions for intrinsic operations by referencing generic operations.



405

406

Figure 2 – Generic operations mappings

407 5.2.2 Recommendations

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

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

413 The following recommendations apply:

- 414 • WBEM protocol mappings and API mappings should define the mapping from a perspective of
 415 the generic operation (i.e., by listing the relevant generic operation at the top level).
- 416 • For each generic operation listed in the mapping, the corresponding WBEM operations or API
 417 calls should be stated that provide the functionality supporting the generic operation.
- 418 • For each parameter defined for a generic operation listed in the mapping, the corresponding
 419 parameters and return values of the WBEM operations or API calls should be stated.
- 420 • A statement should be made for each generic operation as to whether or not the operation is
 421 supported in a conformant way, as defined in 5.3.2. If the operation is supported in a non-
 422 conformant way, the deviations should be stated.
- 423 • A statement should be made for the entire WBEM protocol or API as to whether or not it is
 424 conformant to generic operations.

425 5.3 Conformance to generic operations

426 Conformance to generic operations is defined at two levels:

- 427 1) At the level of the entire WBEM protocol or API

428 2) At the level of single WBEM operations or single API calls

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

432 **5.3.1 Conformance of WBEM protocols or APIs**

433 A WBEM protocol or API is conformant to generic operations if all generic operations defined in this
434 specification are supported by WBEM operations or API calls in a conformant way.

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

437 **5.3.2 Conformance of WBEM operations or API calls**

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

- 440 • The generic operation has one or more corresponding WBEM operations or API calls that
441 provide the functionality of the generic operation. The names of these corresponding WBEM
442 operations or API calls may be different from the name of the generic operation.
- 443 • Functionalities that are required to be supported for a generic operation are supported by the
444 corresponding WBEM operations or API calls with the semantics defined by the generic
445 operation.
- 446 • If functionalities that are optional to be supported for a generic operation are supported by the
447 corresponding WBEM operations or API calls, they are supported with the semantics defined by
448 the generic operation.
- 449 • Each parameter of a generic operation is mapped to one or more corresponding parameters of
450 the corresponding WBEM operations or API calls
- 451 • For each parameter of a generic operation, the provisions defined in 5.3.3 are satisfied.

452 WBEM operations or API calls that support a generic operation in a conformant way, may support
453 parameters or return values in addition to the parameters mapped to parameters of the corresponding
454 generic operation. Defining additional parameters can affect interoperability between WBEM protocols.

455 **5.3.3 Requirement levels for operation parameters**

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

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

460 **Mandatory**

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

465 **Conditional**

466 Operation parameters designated as conditional shall be supported by conformant WBEM
467 protocols or APIs if the specified condition is met. If supported, they shall be supported as

468 defined for the generic operation. Conformant WBEM protocols or APIs may define that
469 supplying values for the corresponding parameters is optional if a default behavior is specified.

470 **Optional**

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

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

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

481 **5.4 Generic types**

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

484 **5.4.1 CIM data types**

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

487 **5.4.2 NamespacePath**

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

489 Conformant WBEM protocols shall support all characteristics of *NamespacePath* values and may support
490 additional characteristics.

491 **5.4.3 InstancePath**

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

493 Conformant WBEM protocols shall support all characteristics of *InstancePath* values and may support
494 additional characteristics.

495 An instance path as defined in [DSP0004](#) allows identifying the name of the creation class of the instance,
496 as well as the names and values of the key properties of the instance.

497 **5.4.4 ClassPath**

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

499 Conformant WBEM protocols shall support all characteristics of *ClassPath* values and may support
500 additional characteristics.

501 **5.4.5 QualifierTypePath**

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

503 Conformant WBEM protocols shall support all characteristics of *ClassPath* values may support additional
504 characteristics.

505 **5.4.6 InstanceSpecification**

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

- 508 • name of the creation class of the instance
- 509 • all or a subset of the static and non-static properties exposed by the creation class of the
510 instance

511 Each property in an *InstanceSpecification* shall contain:

- 512 • name of the property
- 513 • value of the property
- 514 • optional: Class origin of the property
- 515 • optional: Data type of the property

516 *InstanceSpecification* does not contain the instance path of the CIM instance, because there are some
517 situations in which the instance data is needed without an instance path. The
518 *InstanceSpecificationWithPath* type is used when the instance path is needed in addition to the instance
519 data.

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

521 **5.4.7 ClassSpecification**

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

- 524 • name of the class
- 525 • name of the superclass, if any
- 526 • all or a subset of the static and non-static properties (that is, the property definitions) exposed
527 by the class. As defined in [DSP0004](#), the set of properties exposed by a class includes any
528 properties inherited from superclasses, where overridden properties are included only once.
- 529 • all of the static and non-static methods exposed by the class. As defined in [DSP0004](#), the set of
530 methods exposed by a class includes any methods inherited from superclasses, where
531 overridden methods are included only once.
- 532 • optional: all of the qualifiers exposed by the class that are defined on the class or any of its
533 superclasses

534 Each property in a *ClassSpecification* shall contain:

- 535 • name of the property
- 536 • data type of the property
- 537 • default value of the property
- 538 • optional: all of the qualifiers exposed by the property that are defined on the property or any of
539 its overridden properties

540 Each method in a *ClassSpecification* shall contain:

- 541 • name of the method
- 542 • data type of the return value of the method
- 543 • all of the parameters of the method
- 544 • optional: all of the qualifiers exposed by the method that are defined on the method or any of its
- 545 overridden methods

546 Each parameter in that method shall contain:

- 547 • name of the parameter
- 548 • data type of the parameter
- 549 • optional: all of the qualifiers exposed by the parameter that are defined on the parameter or the
- 550 corresponding parameter in any of its overridden methods

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

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

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

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

557 **5.4.8 QualifierType**

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

- 560 • name of the qualifier
- 561 • data type of the qualifier
- 562 • default value of the qualifier
- 563 • all flavors of the qualifier
- 564 • all scopes of the qualifier

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

567 **5.4.9 InstanceSpecificationWithPath**

568 A value of the generic type *InstanceSpecificationWithPath* combines the content of *InstanceSpecification*
569 and *InstancePath*.

570 *InstanceSpecification* shall represent the CIM instance referenced by *InstancePath*.

571 **5.4.10 ClassSpecificationWithPath**

572 A value of the generic type *ClassSpecificationWithPath* combines the content of *ClassSpecification* and
573 *ClassPath*.

574 *ClassSpecification* shall represent the CIM class referenced by *ClassPath*.

575 5.4.11 QualifierTypeWithPath

576 A value of the generic type *QualifierTypeWithPath* combines the content of *QualifierType* and
577 *QualifierTypePath*.

578 *QualifierType* shall represent the CIM qualifier type referenced by *QualifierTypePath*.

579 5.4.12 ClassName

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

581 5.4.13 PropertyName

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

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

584 5.4.14 MethodName

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

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

587 5.4.15 ParameterValue

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

- 590 • name of the parameter
- 591 • value of the parameter
- 592 • optional: Data type of the parameter

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

594 5.4.16 ReturnValue

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

- 597 • return value
- 598 • optional: Data type of the return value

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

600 5.4.17 QueryString

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

603 5.4.18 QueryLanguage

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

605 5.4.19 EnumerationContext

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

608 **5.5 Success and failure**

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

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

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

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

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

618 **5.6 Preconditions and postconditions**

619 Each generic operation specifies a set of zero or more preconditions and a set of zero or more
620 postconditions.

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

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

626 **5.7 Generic error messages**

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

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

- 632 • If the WBEM protocol supports returning DMTF standard messages as part of a failure, then for
633 each of its WBEM operations to which a generic operation was mapped, the WBEM operation
634 shall return the generic error message defined for the generic operation that matches the error
635 situation. The WBEM operation may return additional error messages.
- 636 • If the WBEM protocol supports returning CIM status codes as part of a failure, then for each of
637 its WBEM operations to which a generic operation was mapped, the WBEM operation shall
638 return the CIM status code stated in the generic error message defined for the generic operation
639 that matches the error situation. The CIM status code values are stated in the definition of each
640 message in [DSP8016](#).
- 641 • Otherwise, the WBEM protocol mapping shall state for each of its WBEM operations to which a
642 generic operation was mapped, to which of its protocol specific error conditions each generic
643 error message corresponds that is defined by the generic operation.

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

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

648 Mandatory

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

652 Conditional

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

656 Optional

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

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

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

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

671 5.8 Consistency model

672 This subclause defines consistency requirements for generic operations.

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

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

677 5.8.1 Definition of ACID properties

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

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

684 5.8.1.1 Atomicity

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

687 Atomicity only applies to operations and methods that modify the managed environment or CIM instances
688 through the management interface.

689 5.8.1.2 Update consistency

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

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

693 Update consistency only applies to operations and methods that modify the managed environment or CIM
694 instances through the management interface.

695 5.8.1.3 Isolation

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

699 Isolation applies to operations and methods that retrieve information through the management interface,
700 and to operations that modify the managed environment or CIM instances through the management
701 interface.

702 5.8.1.4 Durability

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

706 Durability only applies to operations and methods that modify the managed environment or CIM instances
707 through the management interface.

708 5.8.2 Time consistency within a CIM instance

709 The property values of an instance returned by any generic operation shall represent a snapshot of the
710 instance in the CIM namespace at some point in time.

711 If a WBEM protocol provides the capability to transfer an operation response in multiple parts, and a
712 response that contains an instance is distributed over multiple parts which are transferred at different
713 points in times, the property values of a particular CIM instance still need to satisfy the time consistency
714 constraint.

715 5.8.3 Staleness of information returned

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

719 5.8.4 Isolation between operations

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

722 For example, if a CIM instance is deleted and after that another one is created, an enumeration operation
723 executed concurrently may consistently include the instance that got deleted just before that happened,
724 as well as the new instance after it got consistently created, hence returning a set of instances that never
725 existed at the same time. This example satisfies all consistency rules defined in this specification.

726 An example where other consistency rules determine the overall behavior is a GetInstance operation
727 executing concurrently with a ModifyInstance operation on the same instance. The consistency rules
728 defined in 5.8.2 require that this GetInstance operation needs to return an instance that either has none
729 or all of the modifications requested by the ModifyInstance operation.

730 **5.8.5 Duplicate return of CIM objects or object paths**

731 Any generic operations returning CIM object specifications or CIM object paths should not return
732 duplicate objects or duplicate object paths.

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

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

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

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

744 **5.8.6 Time consistency between returned CIM objects**

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

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

750 However, the rule defined in 5.8.2 requires that consistency is maintained within each single CIM
751 instance.

752 **5.8.7 Order of returned CIM objects**

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

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

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

760 **5.8.8 Validity of returned object paths**

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

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

765 5.8.9 Effects of deleting an instance

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

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

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

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

776 NOTE: Such dependent instances may reside in a different CIM namespace (which may reside in a
777 different WBEM server) than the instance to be deleted.

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

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

- 782 • **Deletion without propagation:** Delete the instance to be deleted but do not delete any
783 dependent instances. This causes an inconsistent state in the model, so it has not been used
784 for the following types of dependencies.

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

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

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

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

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

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

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

- 799 – by propagating the deletion of the referenced instance to its referencing association
800 instance
- 801 – by rejecting the deletion of the referenced instance to be deleted.

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

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

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

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

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

816 EXAMPLE: Association AB references class A with *Min (2)* and references class B. Therefore, each
817 instance of B is supposed to be associated via AB with least two instances of A. If an instance of A is to
818 be deleted, and there is only one other instance of A associated to the instance of B that is associated
819 with the instance of A to be deleted, the minimum multiplicity would be violated by the deletion.

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

- 821 – by propagating the deletion of the instance to be deleted to its associated instance defining
822 the multiplicity constraint, and the association instance.
- 823 – by rejecting the original deletion.

824 **6 Generic operations**

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

826 **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	See 6.4.1
	EnumerateInstanceNames	See 6.4.2
	Associators	See 6.4.3
	AssociatorNames	See 6.4.4
	References	See 6.4.5
	ReferenceNames	See 6.4.6
Pulled instance enumeration operations	OpenEnumerateInstances	See 6.5.3
	OpenEnumerateInstancePaths	See 6.5.4
	OpenAssociators	See 6.5.5
	OpenAssociatorPaths	See 6.5.6
	OpenReferences	See 6.5.7
	OpenReferencePaths	See 6.5.8
	OpenQueryInstances	See 6.5.9
	PullInstancesWithPath	See 6.5.11
	PullInstancePaths	See 6.5.12
	PullInstances	See 6.5.13
	CloseEnumeration	See 6.5.14
	EnumerationCount	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
	ModifyClass	See 6.7.3
	CreateClass	See 6.7.4

Group	Generic Operation	Description
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

827 **6.1 Description format**

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

830 **Purpose:**

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

832 **Operation Input Parameters:**

833

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>
...

834 **Operation Output Parameters:**

835

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

836 **Description:**

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

839 **Preconditions:**

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

845 **Postconditions:**

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

849 **Error messages:**

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

851

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

- 854 <diname> Generic name of the operation parameter.
- 855 <ditype> Generic type of the operation parameter, as defined in 5.4.
- 856 <direq> Requirement level of the operation parameter, as defined in 5.3.3.
- 857 <msgid> Message ID of the message, as defined in a DMTF message registry. The message
858 ID is the concatenation of the values of the XML attributes
859 MESSAGE/MESSAGE_ID@PREFIX and
860 MESSAGE/MESSAGE_ID@SEQUENCE_NUMBER.
- 861 <msgname> Message name of the message, as defined in a DMTF message registry. The
862 message name is the value of the XML attribute MESSAGE@NAME.
- 863 <msgreq> Requirement level of the message, as defined in 5.7.
- 864 <msgsrc> Sources of the message. One or more values may be specified. Valid values are:
- 865 Infrastructure – the message is implemented by the common infrastructure portion
866 of the WBEM server.
- 867 Class implem. – the message is implemented by the class specific portion of the
868 WBEM server.
- 869 The message sources information is a recommendation only, for implementations of
870 a WBEM server that distinguish between a common infrastructure portion (e.g.,
871 CIMOM) and class specific portion (e.g., providers).

872 **6.2 Common operation parameters for all operations**

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

876 **6.2.1 IncludeClassOrigin**

877 The *IncludeClassOrigin* operation input parameter controls whether class origin information is returned for
878 any element in any returned object. Class origin information indicates which class defines the element.

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

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

884 If the WBEM protocol supports client side control of returning class origin information, then the
885 *IncludeClassOrigin* operation parameter shall be supported. If the *IncludeClassOrigin* operation
886 parameter is TRUE, then class origin information shall be included for any element in any object returned
887 by the operation. If the *IncludeClassOrigin* operation parameter is FALSE, then class origin information
888 shall not be included for any element in any object returned by the operation.

889 For operations returning instances, the elements are properties only (more precisely, their values). For
890 operations returning classes, the elements are properties and methods (more precisely, their definitions).

891 **6.2.2 IncludeQualifiers**

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

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

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

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

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

903 **6.2.3 <element>List**

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

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

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

911 6.3 Instance operations

912 This subclause defines instance operations (operations that target a single CIM instance, or create a CIM
913 instance).

914 6.3.1 GetInstance

915 Purpose:

916 Retrieves a CIM instance.

917 Operation Input Parameters:

918

Generic Name	Generic Type	Requirement	Description
InstancePath	InstancePath	Mandatory	Instance path of the instance to be retrieved (Context Parameter)
IncludeClassOrigin	boolean	Conditional	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

919 Operation Output Parameters:

920

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

921 Description:

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

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

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

- 928 • Initially, the set of properties to be included is the set of properties exposed by the creation
929 class of the instance. This includes all the duplicates of any duplicate non-overridden
930 properties.
- 931 • If the *IncludedProperties* operation input parameter is supported by the WBEM protocol
932 and if its value is not NULL, it acts as a restricting filter on the properties to be included in
933 the returned instance such that any properties exposed by the creation class of the
934 instance that are not named in that operation parameter are removed from the set of
935 properties to be included. Any duplicate or invalid property names in the
936 *IncludedProperties* operation input parameter shall be ignored. A non-NULL empty
937 *IncludedProperties* list removes all properties from the set of properties to be included.

- 938 • Conformant WBEM protocols may specify rules that cause properties with a value of NULL
939 to be removed from the set of properties to be included.

940 **Preconditions:**

- 941 • The instance referenced by *InstancePath* shall exist in the namespace. If this is not satisfied,
942 the operation shall fail, indicating WIPG0213.

943 **Postconditions:**

- 944 • The instance shall have been returned with the properties as defined in the Description
945 paragraph for this operation.
- 946 • Requirements on ACID properties:
- 947 – Atomicity: N/A
- 948 – Update Consistency: N/A
- 949 – Isolation: Required
- 950 – Durability: N/A

951 **Error messages:**

952

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM service 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.	
WIPG0213	Instance not found	Mandatory	Class implem.	
WIPG0243	Timeout	Optional	Infrastructure, class implem.	
WIPG0227	Other failure	Optional	Infrastructure, class implem.	

953 **6.3.2 DeleteInstance**954 **Purpose:**

955 Deletes a CIM instance.

956 **Operation Input Parameters:**

957

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

958 **Operation Output Parameters:**

959 None.

960 **Description:**961 The *DeleteInstance* operation deletes the instance referenced by *InstancePath*.

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

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

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

971 The effects of the deletion of any CIM instances on any underlying resources shall be defined
 972 elsewhere. For example, a management profile may define that the lifecycle of the CIM instance is
 973 coupled with the lifecycle of some underlying resource, and that this resource shall be deleted when
 974 the instance is deleted.

975 **Preconditions:**

- 976 • The instance referenced by *InstancePath* shall exist in the namespace. If this is not satisfied,
 977 the operation shall fail, indicating WIPG0213.

978 **Postconditions:**

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

- 989 – Update Consistency: Required, if dependent instances are handled by rejection, as defined
- 990 in 5.8.9. Recommended, if dependent instances are handled by delete propagation, as
- 991 defined in 5.8.9.
- 992 – Isolation: Required, if dependent instances are handled by rejection, as defined in 5.8.9.
- 993 Recommended, if dependent instances are handled by delete propagation, as defined in
- 994 5.8.9.
- 995 – Durability: Required.

996 **Error messages:**

997

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM service 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.	
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.	

998 **6.3.3 ModifyInstance**999 **Purpose:**

1000 Changes property values of a CIM 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 **Operation Output Parameters:**

1004 None.

1005 **Description:**1006 The *ModifyInstance* operation changes property values of the instance referenced by *InstancePath*.

1007 The set of properties to be changed shall be determined using the following algorithm:

- 1008 • Initially, the set of properties to be changed is the set of properties specified in
1009 *ModifiedInstance*.
- 1010 • If the *IncludedProperties* operation input parameter is supported by the WBEM protocol
1011 and if its value is not NULL, it acts as a restricting filter on the properties to be changed
1012 such that any properties exposed by the creation class of the instance that are not named
1013 in that operation parameter are removed from the set of properties to be changed. Any
1014 duplicate or invalid property names in the *IncludedProperties* operation input parameter
1015 shall be ignored. A non-NULL empty *IncludedProperties* list removes all properties from
1016 that set.
- 1017 • Any key properties and non-modifiable properties are removed from the set of properties to
1018 be changed. As a result, specifying such properties in *ModifiedInstance* or
1019 *IncludedProperties* does not cause an error.

1020 NOTE: The modifiability of properties can be defined in the schema and in management profiles.

1021 Conformant WBEM protocols may restrict *ModifiedInstance* to specify all properties exposed by the
1022 creation class of the instance referenced by *InstancePath*.1023 **Preconditions:**

- 1024 • The instance referenced by *InstancePath* shall exist in the namespace. If this is not satisfied,
1025 the operation shall fail, indicating WIPG0213.
- 1026 • The creation class of *ModifiedInstance* shall be the creation class of the instance referenced by
1027 *InstancePath* or a superclass of that class. If this is not satisfied, the operation shall fail,
1028 indicating WIPG0208.

- 1029 • Any properties specified in *ModifiedInstance* shall be from the set of properties exposed by the
- 1030 creation class of *ModifiedInstance*. If this is not satisfied, the operation shall fail, indicating
- 1031 WIPG0208.

1032 **Postconditions:**

- 1033 • The values of the properties shall have been modified as defined in the Description paragraph
- 1034 for this operation.
- 1035 • The values of key properties and non-modifiable properties shall not have been modified.
- 1036 • Other properties may have changed as a result of side effects of changing properties, behavior
- 1037 defined in referencing specifications, or volatility of properties.
- 1038 • The consistency requirements defined in [DSP0004](#) shall be satisfied for the modified instance.
- 1039 • Requirements on ACID properties:
 - 1040 – Atomicity: Required
 - 1041 – Update Consistency: Required
 - 1042 – Isolation: Required
 - 1043 – Durability: Required

1044 **Error messages:**

1045

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM service 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.	

1046 **6.3.4 CreateInstance**1047 **Purpose:**

1048 Creates a CIM instance.

1049 **Operation Input Parameters:**

1050

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

1051 **Operation Output Parameters:**

1052

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

1053 **Description:**

1054 The *CreateInstance* operation creates a CIM instance in the namespace specified in *ClassPath* from
 1055 the creation class specified in *ClassPath*, and returns the instance path of the new instance.

1056 As defined in the description of the *ClassPath* type, the class path of the CIM class to be used as a
 1057 creation class for the instance is interpreted in a non-polymorphic way, i.e., it references the
 1058 specified class only and not any subclasses. In other words, the instance is created from the
 1059 specified class only. As a result, the specified class becomes the creation class of the instance.

1060 The newly created instance shall have all properties exposed by the creation class specified in
 1061 *ClassPath*.

1062 For each property, the initial value shall be determined as defined in the following **default** rules:

- 1063 • If the *NewInstance* operation input parameter is supported, and if the property is included
 1064 in *NewInstance*, its value is used. That is also the case if that value is NULL.
- 1065 • Else, if a default value is declared for the property, that value is used.

1066 These default rules allow specifying key properties and non-writeable properties in *NewInstance*. In
 1067 other words, the creation of an instance does not have the restrictions a subsequent modification
 1068 has.

1069 As defined in [DSP1001](#), management profiles may specify any such rules, overriding these default
 1070 rules. This may result in rejecting, respecting or replacing the values of any properties specified in
 1071 *NewInstance*, as well as respecting or replacing the default values of any properties not specified in
 1072 *NewInstance*.

1073 Volatile properties may change their values immediately after the instance has been created.

1074 Instance creation based upon input data other than initial property values can be done using CIM
 1075 methods. For example, creation of an instance of *CIM_ComputerSystem* representing a virtual
 1076 computer system could be done using a *CreateVirtualComputerSystem()* method taking a higher-
 1077 level specification of the virtual computer system as input.

1078 Other CIM instances may come into existence implicitly during the course of processing the
 1079 *CreateInstance* operation. As defined in [DSP1001](#), management profiles may specify the rules for
 1080 such implicitly created instances.

1081 Any such implicitly created instances may reside in a different CIM namespace (which may reside in
 1082 a different WBEM server) than the namespace specified in *ClassPath*.

1083 In case of error, the consistency requirements defined in [DSP0004](#) should be attempted to be
 1084 satisfied in a best effort approach. In case of error, only a subset of the instances to be created may
 1085 have been created, but each instance shall have either been created completely or not at all.

1086 As defined in [DSP1001](#), management profiles may specify the effects of the creation of CIM
 1087 instances on their underlying resources. For example, a management profile may define that the
 1088 lifecycle of the CIM instance is coupled with the lifecycle of some underlying resource, and that this
 1089 resource shall be created when the instance is created.

1090 **Preconditions:**

- 1091 • The instance to be created shall not exist in the namespace specified by *ClassPath*. If this is not
 1092 satisfied, the operation shall fail, indicating WIPG0216.
- 1093 • The CIM class referenced by *ClassPath* shall exist in the namespace. If this is not satisfied, the
 1094 operation shall fail, indicating WIPG0214.
- 1095 • The creation class of *NewInstance* shall be the class referenced by *ClassPath* or a superclass
 1096 of that class. If this is not satisfied, the operation shall fail, indicating WIPG0208.
- 1097 • Any properties specified in *NewInstance* shall be from the set of properties exposed by the
 1098 class referenced by *ClassPath*. If this is not satisfied, the operation shall fail, indicating
 1099 WIPG0208.

1100 **Postconditions:**

- 1101 • The instance shall have been created as defined in the Description paragraph for this operation.
- 1102 • Any management profile defined implicit creations of other CIM instances shall have happened.
- 1103 • Any management profile defined effects of the creation of all of these CIM instances on any
 1104 underlying resources shall have happened.
- 1105 • Requirements on ACID properties:
 - 1106 – Atomicity: Required
 - 1107 – Update Consistency: Required
 - 1108 – Isolation: Required
 - 1109 – Durability: Required

1110 **Error messages:**

1111

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM service 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, 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.	

1112 **6.4 Direct instance enumeration operations**

1113 This subclause defines direct instance enumeration operations (operations that enumerate CIM instances
1114 and return them directly as a result of the operation).

1115 **6.4.1 EnumerateInstances**

1116 **Purpose:**

1117 Enumerate the CIM instances of a class and return these instances.

1118 **Operation Input Parameters:**

1119

Generic Name	Generic Type	Requirement	Description
EnumClassPath	ClassPath	Mandatory	Class path of CIM class used for the enumeration (Context Parameter)
IncludeClassOrigin	boolean	Conditional	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
ExcludeSubclass-Properties	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 instances

1120 **Operation Output Parameters:**

1121

Generic Name	Generic Type	Requirement	Description
InstanceList	InstanceSpecificationWithPath []	Mandatory	Sequence of the enumerated instances with their instance paths

1122 **Description:**

1123 The *EnumerateInstances* operation enumerates all CIM instances of the class referenced by
1124 *EnumClassPath*, including instances of any of its subclasses, and returns these instances together
1125 with their instance paths.

1126 All of the instances returned shall exist in the same namespace as the class referenced by
1127 *EnumClassPath*.

1128 An instance is included in the result set if and only if it exists in the namespace specified in
1129 *EnumClassPath*, and its creation class is the class specified in *EnumClassPath* or a subclass of that
1130 class.

1131 The result set should not contain any duplicate instances, as defined in 5.8.4. Because the result set
1132 contains only instances that exist in the same namespace, a determination of duplicate instances (for
1133 example by the Client) can be done on the basis of their model paths only.

- 1134 The set of properties to be included in any instances in the result set shall be determined using the
1135 following algorithm:
- 1136 • Initially, the set of properties to be included is the set of properties exposed by the creation
1137 class of the instance. This includes all the duplicates of any duplicate non-overridden
1138 properties.
 - 1139 • If the *IncludedProperties* operation input parameter is supported by the WBEM protocol
1140 and if its value is not NULL, it acts as a restricting filter on the properties to be included in
1141 the returned instances such that any properties exposed by the creation class of the
1142 instance that are not named in that operation parameter are removed from the set of
1143 properties to be included. Any duplicate or invalid property names in the
1144 *IncludedProperties* operation input parameter shall be ignored. A non-NULL empty
1145 *IncludedProperties* list removes all properties from the set of properties to be included.
 - 1146 • If the *ExcludeSubclassProperties* operation input parameter is supported by the WBEM
1147 protocol and if its value is TRUE, it acts as a restricting filter on the properties to be
1148 included in the returned instances such that any properties not exposed by the class
1149 referenced by *EnumClassPath* are removed from the set of properties to be included. In
1150 other words, the set of properties is restricted to the properties exposed by the
1151 enumeration class.
 - 1152 • Conformant WBEM protocols may specify rules that cause properties with a value of NULL
1153 to be removed from the set of properties to be included.

1154 **Preconditions:**

- 1155 • The CIM class referenced by *EnumClassPath* shall exist in the namespace. If this is not
1156 satisfied, the operation shall fail, indicating WIPG0214.

1157 **Postconditions:**

- 1158 • The enumerated instances with their instance paths shall have been returned as described in
1159 the Description paragraph for this operation.
- 1160 • Requirements on ACID properties:
 - 1161 – Atomicity: N/A
 - 1162 – Update Consistency: N/A
 - 1163 – Isolation: Required at the level of single instances, as defined in 5.8.
 - 1164 – Durability: N/A

1165 **Error Messages:**

1166

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

1167 **6.4.2 EnumerateInstanceNames**

1168 **Purpose:**

1169 Enumerate the CIM instances of a class and return their instance paths.

1170 **Operation Input Parameters:**

1171

Generic Name	Generic Type	Requirement	Description
EnumClassPath	ClassPath	Mandatory	Class path of CIM class used for the enumeration (Context Parameter)

1172 **Operation Output Parameters:**

1173

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

1174 **Description:**

1175 The *EnumerateInstanceNames* operation enumerates all CIM instances of the class referenced by
1176 *EnumClassPath*, and returns the instance paths of these instances.

1177 An instance is included in the result set if and only if it exists in the namespace specified in
1178 *EnumClassPath*, and its creation class is the class specified in *EnumClassPath* or a subclass of that
1179 class.

1180 The result set should not contain any duplicate instances, as defined in 5.8.4. Because the result set
1181 contains only instances that exist in the same namespace, a determination of duplicate instances (for
1182 example by the Client) can be done on the basis of their model paths only.

1183 **Preconditions:**

- 1184 • The CIM class referenced by *EnumClassPath* shall exist in the namespace. If this is not
1185 satisfied, the operation shall fail, indicating WIPG0214.

1186 **Postconditions:**

- 1187 • The instance paths of the enumerated instances shall have been returned as described in the
1188 Description paragraph for this operation.
- 1189 • Requirements on ACID properties:
- 1190 – Atomicity: N/A
- 1191 – Update Consistency: N/A
- 1192 – Isolation: Required at the level of single instances, as defined in 5.8.
- 1193 – Durability: N/A

1194 **Error messages:**

1195

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

1196 **6.4.3 Associators**

1197 **Purpose:**

1198 Enumerate CIM instances that are associated with a given source instance and return those
 1199 instances together with their instance paths.

1200 **Operation Input Parameters:**
 1201

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

1202 **Operation Output Parameters:**
 1203

Generic Name	Generic Type	Requirement	Description
InstanceList	InstanceSpecificationWithPath []	Mandatory	Sequence of the associated instances with their instance paths

1204 **Description:**

1205 The *Associators* operation enumerates instances that are associated with a given source instance
 1206 and returns these instances together with their instance paths.

- 1207 The set of associated instances to be returned shall be determined using the following algorithm:
- 1208 • Initially, the set of instances to be returned is the set of all instances associated to the
1209 source instance specified in *SourceInstancePath*. The associations may be instances of
1210 different association classes.
 - 1211 The result set should not contain any duplicate instances, as defined in 5.8.4. However,
1212 different far ends may reference the same instance, and in such cases, the instance shall
1213 be contained in the result set once for each such reference.
 - 1214 • If the *AssociationClassName* operation input parameter is not NULL, it acts as a restricting
1215 filter on the instances to be returned such that each instance that is associated with the
1216 source instance using an association whose creation class or one of its superclasses does
1217 not have the name specified in *AssociationClassName*, is removed from the set of
1218 instances to be returned. There shall be no validity checking performed for the
1219 *AssociationClassName* operation input parameter.
 - 1220 • If the *AssociatedClassName* operation input parameter is not NULL, it acts as a restricting
1221 filter on the instances to be returned such that each instance whose creation class or one
1222 of its superclasses does not have the name specified in *AssociatedClassName*, is removed
1223 from the set of instances to be returned. There shall be no validity checking performed for
1224 the *AssociatedClassName* operation input parameter.
 - 1225 • If the *SourceRoleName* operation input parameter is not NULL, it acts as a restricting filter
1226 on the instances to be returned such that each instance that is associated with the source
1227 instance using an association class that has a role name on the source end that is not the
1228 role name specified in *SourceRoleName*, is removed from the set of instances to be
1229 returned. There shall be no validity checking performed for the *SourceRoleName* operation
1230 input parameter.
 - 1231 • If the *AssociatedRoleName* operation input parameter is not NULL, it acts as a restricting
1232 filter on the instances to be returned such that each instance that is associated with the
1233 source instance using an association class that has a role name on the end referencing
1234 that instance that is not the role name specified in *AssociatedRoleName*, is removed from
1235 the set of instances to be returned. There shall be no validity checking performed for the
1236 *AssociatedRoleName* operation input parameter.
- 1237 The set of properties to be included in each returned associated instance shall be determined using
1238 the following algorithm:
- 1239 • Initially, the set of properties to be included is the set of properties exposed by the creation
1240 class of the instance. This includes all the duplicates of any duplicate non-overridden
1241 properties.
 - 1242 • If the *IncludedProperties* operation input parameter is supported by the WBEM protocol
1243 and if its value is not NULL, it acts as a restricting filter on the properties to be included in
1244 the returned instances such that any properties exposed by the creation class of the
1245 instance that are not named in that operation parameter are removed from the set of
1246 properties to be included. Any duplicate or invalid property names in the
1247 *IncludedProperties* operation input parameter shall be ignored. A non-NULL empty
1248 *IncludedProperties* list removes all properties from the set of properties to be included.
 - 1249 • If the *ExcludeSubclassProperties* operation input parameter is supported by the WBEM
1250 protocol and if its value is TRUE, it acts as a restricting filter on the properties to be
1251 included in the returned instances such that any properties not exposed by the class
1252 specified in *AssociatedClassName* are removed from the set of properties to be included.
 - 1253 • Conformant WBEM protocols may specify rules that cause properties with a value of NULL
1254 to be removed from the set of properties to be included.

1255 **Preconditions:**

- 1256 • The instance referenced by *SourceInstancePath* shall exist in the namespace. If this is not
1257 satisfied, the operation shall fail, indicating WIPG0213.
- 1258 • The *IncludedProperties* operation parameter, if supported by the WBEM protocol, shall only be
1259 specified with a non-NULL value if the *AssociatedClassName* operation input parameter is also
1260 non-NULL. If this is not satisfied, the operation shall fail, indicating WIPG0208.
- 1261 • The *ExcludeSubclassProperties* operation parameter, if supported by the WBEM protocol, shall
1262 only be specified with a TRUE value if the *AssociatedClassName* operation input parameter is
1263 non-NULL. If this is not satisfied, the operation shall fail, indicating WIPG0208.

1264 NOTE: Specifying a non-NULL value for *AssociatedClassName* ensures that the associated instances have the
1265 class specified in *AssociatedClassName* as a common superclass.

1266 **Postconditions:**

- 1267 • The associated instances with their instance paths shall have been returned as described in the
1268 Description paragraph for this operation.
- 1269 • Requirements on ACID properties:
 - 1270 – Atomicity: N/A
 - 1271 – Update Consistency: N/A
 - 1272 – Isolation: Required at the level of single instances, as defined in 5.8.
 - 1273 – Durability: N/A

1274 **Error Messages:**

1275

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0240	WBEM service limits are exceeded	Optional	Infrastructure, class implem.	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM service 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.	
WIPG0213	Instance not found	Mandatory	Class implem.	

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

1276 6.4.4 AssociatorNames

1277 Purpose:

1278 Enumerate CIM instances that are associated with a given source instance and return their instance
1279 paths.

1280 Operation Input Parameters:

1281

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

1282 Operation Output Parameters:

1283

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

1284 Description:

1285 The *AssociatorNames* operation enumerates the instance paths of instances that are associated with
1286 a given source instance and returns these instance paths.

1287 The set of associated instances of which instance paths are to be returned shall be determined using
1288 the following algorithm:

- 1289 • Initially, the set of instances to be returned is the set of all instances associated to the
1290 source instance specified in *SourceInstancePath*. The associations may be instances of
1291 different association classes.

1292 The result set should not contain any duplicate instances, as defined in 5.8.4. However,
1293 different association instances may reference the same instance on one of their far ends,

- 1294 and in such cases, the instance shall be contained in the result set once for each such
1295 reference.
- 1296 • If the *AssociationClassName* operation input parameter is not NULL, it acts as a restricting
1297 filter on the instances to be returned such that each instance that is associated with the
1298 source instance using an association whose creation class or one of its superclasses does
1299 not have the name specified in *AssociationClassName*, is removed from the set of
1300 instances to be returned. There shall be no validity checking performed for the
1301 *AssociationClassName* operation input parameter.
 - 1302 • If the *AssociatedClassName* operation input parameter is not NULL, it acts as a restricting
1303 filter on the instances to be returned such that each instance whose creation class or one
1304 of its superclasses does not have the name specified in *AssociatedClassName*, is removed
1305 from the set of instances to be returned. There shall be no validity checking performed for
1306 the *AssociatedClassName* operation input parameter.
 - 1307 • If the *SourceRoleName* operation input parameter is not NULL, it acts as a restricting filter
1308 on the instances to be returned such that each instance that is associated with the source
1309 instance using an association class that has a role name on the source end that is not the
1310 role name specified in *SourceRoleName*, is removed from the set of instances to be
1311 returned. There shall be no validity checking performed for the *SourceRoleName* operation
1312 input parameter.
 - 1313 • If the *AssociatedRoleName* operation input parameter is not NULL, it acts as a restricting
1314 filter on the instances to be returned such that each instance that is associated with the
1315 source instance using an association class that has a role name on the end referencing
1316 that instance that is not the role name specified in *AssociatedRoleName*, is removed from
1317 the set of instances to be returned. There shall be no validity checking performed for the
1318 *AssociatedRoleName* operation input parameter.

1319 The consistency model defined in 5.8 applies.

1320 **Preconditions:**

- 1321 • The instance referenced by *SourceInstancePath* shall exist in the namespace. If this is not
1322 satisfied, the operation shall fail, indicating WIPG0213.

1323 **Postconditions:**

- 1324 • The instance paths of the associated instances shall have been returned as described in the
1325 Description paragraph for this operation.
- 1326 • Requirements on ACID properties:
 - 1327 – Atomicity: N/A
 - 1328 – Update Consistency: N/A
 - 1329 – Isolation: Required at the level of single instances, as defined in 5.8.
 - 1330 – Durability: N/A

1331
1332**Error Messages:**

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0240	WBEM service limits are exceeded	Optional	Infrastructure, class implem.	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM service 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.	
WIPG0213	Instance not found	Mandatory	Class implem.	
WIPG0243	Timeout	Optional	Infrastructure, class implem.	
WIPG0227	Other failure	Optional	Infrastructure, class implem.	

1333 **6.4.5 References**

1334 **Purpose:**

1335 Enumerate CIM association instances that reference a given source instance and return these
 1336 instances together with their instance path.

1337 **Operation Input Parameters:**

1338

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

1339 **Operation Output Parameters:**

1340

Generic Name	Generic Type	Requirement	Description
InstanceList	InstanceSpecificationWithPath []	Mandatory	Sequence of the association instances with their instance paths

1341 **Description:**

1342 The *References* operation enumerates association instances that reference the specified source
 1343 instance and returns these instances together with their instance paths.

- 1344 The set of association instances to be returned shall be determined using the following algorithm:
- 1345 • Initially, the set of instances to be returned is the set of all instances referencing the source
1346 instance specified in *SourceInstancePath*. These associations may be instances of
1347 different association classes.
 - 1348 • If the *AssociationClassName* operation input parameter is not NULL, it acts as a restricting
1349 filter on the instances to be returned such that each association instance whose creation
1350 class or one of its superclasses does not have the name specified in
1351 *AssociationClassName*, is removed from the set of instances to be returned. There shall be
1352 no validity checking performed for the *AssociationClassName* operation input parameter.
 - 1353 • If the *AssociatedClassName* operation input parameter is not NULL, it acts as a restricting
1354 filter on the instances to be returned such that each association instance whose creation
1355 class has a set of far ends none of which is referencing a class where that class or one of
1356 its superclasses has the name specified in *AssociatedClassName*, is removed from the set
1357 of instances to be returned. There shall be no validity checking performed for the
1358 *AssociatedClassName* operation input parameter.
 - 1359 • If the *SourceRoleName* operation input parameter is not NULL, it acts as a restricting filter
1360 on the instances to be returned such that each association instance whose creation class
1361 does not have the role name specified in *SourceRoleName* on the end referencing the
1362 source instance, is removed from the set of instances to be returned. There shall be no
1363 validity checking performed for the *SourceRoleName* operation input parameter.
 - 1364 • If the *AssociatedRoleName* operation input parameter is not NULL, it acts as a restricting
1365 filter on the instances to be returned such that each association instance whose creation
1366 class has a set of far ends none of which has the role name specified in
1367 *AssociatedRoleName*, is removed from the set of instances to be returned. There shall be
1368 no validity checking performed for the *AssociatedRoleName* operation input parameter.
- 1369 The consistency model defined in 5.8 applies.
- 1370 The set of properties to be included in each returned association instance shall be determined using
1371 the following algorithm:
- 1372 • Initially, the set of properties to be included is the set of properties exposed by the creation
1373 class of the instance. This includes all the duplicates of any duplicate non-overridden
1374 properties.
 - 1375 • If the *IncludedProperties* operation input parameter is supported by the WBEM protocol
1376 and if its value is not NULL, it acts as a restricting filter on the properties to be included in
1377 the returned instances such that any properties exposed by the creation class of the
1378 instance that are not named in that operation parameter are removed from the set of
1379 properties to be included. Any duplicate or invalid property names in the
1380 *IncludedProperties* operation input parameter shall be ignored. A non-NULL empty
1381 *IncludedProperties* list removes all properties from the set of properties to be included
 - 1382 • If the *ExcludeSubclassProperties* operation input parameter is supported by the WBEM
1383 protocol and if its value is TRUE, it acts as a restricting filter on the properties to be
1384 included in the returned instances such that any properties not exposed by the class
1385 specified in *AssociationClassName* are removed from the set of properties to be included.
 - 1386 • Conformant WBEM protocols may specify rules that cause properties with a value of NULL
1387 to be removed from the set of properties to be included.

1388 **Preconditions:**

- 1389 • The instance referenced by *SourceInstancePath* shall exist in the namespace. If this is not
1390 satisfied, the operation shall fail, indicating WIPG0213.
- 1391 • The *IncludedProperties* operation parameter, if supported by the WBEM protocol, shall only be
1392 specified with a non-NULL value if the *AssociationClassName* operation input parameter is also
1393 non-NULL. If this is not satisfied, the operation shall fail, indicating WIPG0208.
- 1394 • The *ExcludeSubclassProperties* operation parameter, if supported by the WBEM protocol, shall
1395 only be specified with a TRUE value if the *AssociationClassName* operation input parameter is
1396 non-NULL. If this is not satisfied, the operation shall fail, indicating WIPG0208.

1397 NOTE: Specifying a non-NULL value for *AssociationClassName* ensures that the association instances have the
1398 class specified in *AssociationClassName* as a common superclass.

1399 **Postconditions:**

- 1400 • The association instances with their instance paths shall have been returned as described in the
1401 Description paragraph for this operation.
- 1402 • Requirements on ACID properties:
 - 1403 – Atomicity: N/A
 - 1404 – Update Consistency: N/A
 - 1405 – Isolation: Required at the level of single instances, as defined in 5.8.
 - 1406 – Durability: N/A

1407 **Error Messages:**

1408

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0240	WBEM service limits are exceeded	Optional	Infrastructure, class implem.	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM service 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.	
WIPG0213	Instance not found	Mandatory	Class implem.	

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

1409 6.4.6 ReferenceNames

1410 Purpose:

1411 Enumerate CIM association instances that reference a given source instance and return their
1412 instance paths.

1413 Operation Input Parameters:

1414

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

1415 Operation Output Parameters:

1416

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

1417 Description:

1418 The *ReferenceNames* operation enumerates the instance paths of association instances that
1419 reference the specified source instance and returns these instance paths.

1420 The set of association instances of which instance paths are to be returned shall be determined
1421 using the following algorithm:

- 1422 • Initially, the set of instances to be returned is the set of all instances referencing the source
1423 instance specified in *SourceInstancePath*. These associations may be instances of
1424 different association classes.
- 1425 • If the *AssociationClassName* operation input parameter is not NULL, it acts as a restricting
1426 filter on the instances to be returned such that each association instance whose creation

- 1427 class or one of its superclasses does not have the name specified in
 1428 *AssociationClassName*, is removed from the set of instances to be returned. There shall be
 1429 no validity checking performed for the *AssociationClassName* operation input parameter.
- 1430 • If the *AssociatedClassName* operation input parameter is not NULL, it acts as a restricting
 1431 filter on the instances to be returned such that each association instance whose creation
 1432 class has a set of far ends none of which is referencing a class where that class or one of
 1433 its superclasses has the name specified in *AssociatedClassName*, is removed from the set
 1434 of instances to be returned. There shall be no validity checking performed for the
 1435 *AssociatedClassName* operation input parameter.
 - 1436 • If the *SourceRoleName* operation input parameter is not NULL, it acts as a restricting filter
 1437 on the instances to be returned such that each association instance whose creation class
 1438 does not have the role name specified in *SourceRoleName* on the end referencing the
 1439 source instance, is removed from the set of instances to be returned. There shall be no
 1440 validity checking performed for the *SourceRoleName* operation input parameter.
 - 1441 • If the *AssociatedRoleName* operation input parameter is not NULL, it acts as a restricting
 1442 filter on the instances to be returned such that each association instance whose creation
 1443 class has a set of far ends none of which has the role name specified in
 1444 *AssociatedRoleName*, is removed from the set of instances to be returned. There shall be
 1445 no validity checking performed for the *AssociatedRoleName* operation input parameter.

1446 The consistency model defined in 5.8 applies.

1447 **Preconditions:**

- 1448 • The instance referenced by *SourceInstancePath* shall exist in the namespace. If this is not
 1449 satisfied, the operation shall fail, indicating WIPG0213.

1450 **Postconditions:**

- 1451 • The instance paths of the association instances shall have been returned as described in the
 1452 Description paragraph for this operation.
- 1453 • Requirements on ACID properties:
 - 1454 – Atomicity: N/A
 - 1455 – Update Consistency: N/A
 - 1456 – Isolation: Required at the level of single instances, as defined in 5.8.
 - 1457 – Durability: N/A

1458 **Error Messages:**

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

1460 **6.5 Pulled instance enumeration operations**

1461 This subclause defines pulled instance enumeration operations (operations that enumerate CIM
1462 instances and return them by means of subsequent pull operations).

1463 The common pattern for these operations is that an enumeration session gets established through an
1464 "Open" operation, also establishing the kind of operation and the kind of items to be returned (instances
1465 or instance paths of instances), and subsequent repeated executions of a "Pull" operation on the
1466 enumeration session are used to retrieve the items. Optionally, the "Open" operation can also pull a first
1467 set of items.

1468 The pulled instance enumeration operations consist of the following individual operations:

1469 • Open operations:

1470 OpenEnumerateInstances – Open an enumeration of instances of a class

1471 OpenEnumerateInstancePaths – Open an enumeration of the instance paths of instances of a
1472 class

1473 OpenAssociators – Open an enumeration of instances associated to a source instance

1474 OpenAssociatorPaths – Open an enumeration of the instance paths of instances associated to
1475 a source instance

1476 OpenReferences – Open an enumeration of association instances referencing a source
1477 instance

1478 OpenReferencePaths – Open an enumeration of the instance paths of association instances
1479 referencing a source instance

1480 OpenQueryInstances – Open an enumeration of instances representing a query result

1481 • Pull operations:

1482 PullInstancesWithPath – Pull operation for retrieving instances with paths

1483 PullInstancePaths – Pull operation for retrieving instance paths

1484 PullInstances – Pull operation for retrieving instances without paths representing query results

1485 • Other operations:

1486 CloseEnumeration – Close an open enumeration

1487 EnumerationCount – Estimate number of items in an open enumeration

1488 **6.5.1 General behavioral rules**

1489 A central concept of the pulled instance enumeration operations is the "enumeration session". An
1490 enumeration session can be thought of as a context in which the operations perform their work, and
1491 which determines the set of objects to be returned. In order to process the operations related to an
1492 enumeration session, some of the operation parameters of the Open operation need to be maintained as
1493 long as the enumeration session is open, as well as some state data about where the enumeration
1494 session is with respect to objects already returned.

1495 From a WBEM client's perspective, an enumeration session is represented as an enumeration context
1496 value. A successful Open operation establishes the enumeration session and returns an enumeration
1497 context value representing the open enumeration session. The enumeration context value is used as an
1498 operation input/output parameter in subsequent Pull operations on that enumeration session. The

1499 enumeration context value shall uniquely identify the open enumeration session within the target CIM
1500 namespace of the Open operation that established the enumeration session. This does not require the
1501 enumeration context value to be time-unique, i.e., it may be reused for a new enumeration session after
1502 the old enumeration session was closed. It is valid for a WBEM server to use NULL as an enumeration
1503 context value representing a closed enumeration session, but a WBEM client shall not rely on that to
1504 detect that an enumeration session has been closed.

1505 Defining the enumeration context value in Pull operations not only as an operation input parameter but
1506 also as an operation output parameter allows the WBEM server to change the enumeration context value
1507 during the execution of a Pull operation. This allows for different implementation approaches for the
1508 WBEM server, which are transparent for the WBEM client.

1509 Example approaches are:

- 1510 • maintaining any state data describing the enumeration session internally in the WBEM server.
1511 In this approach, the enumeration context value does not need to change in subsequent Pull
1512 operations. It is used by the WBEM server only to identify the internal state data for the open
1513 enumeration session, but it is not used to store any of the state data in it. A variation of this
1514 approach is to hand back modified enumeration context values for additional WBEM server side
1515 sequence checking.
- 1516 • maintaining any state data describing the enumeration session on the WBEM client side only. In
1517 this approach, all state data is stored in the enumeration context value, and the WBEM server
1518 does not maintain any state data about the enumeration session, essentially being completely
1519 stateless with respect to the enumeration session.
- 1520 • a combination of the two previous approaches

1521 A WBEM server may support keeping enumeration sessions open across connection terminations and
1522 shutdowns of the server. Objects may be created, deleted or modified concurrently with an enumeration
1523 session that involves these objects. Such changes may or may not be reflected in the enumeration set.
1524 Therefore, there is no guarantee to the WBEM client that the enumeration set represents a consistent
1525 snapshot of its objects at a point in time. However, the WBEM server should make a best effort attempt
1526 for the returned enumeration set to represent a consistent snapshot of its objects at a point in time. The
1527 order of objects in the enumeration set is undefined.

1528 This specification does not define any restrictions on the number of enumeration sessions that can be
1529 established or executed on concurrently in the same WBEM server or by the same WBEM client. This
1530 remains true even if the enumeration sets of such concurrently established enumeration sessions contain
1531 the same objects.

1532 With the exception of CloseEnumeration, all operations on a particular enumeration session shall be
1533 executed sequentially. An enumeration session can be open or closed. The enumeration session is
1534 considered open if operations using its enumeration context value as an operation input parameter can
1535 be executed successfully. It is opened by the successful completion of an Open operation and closed by
1536 one of the following:

- 1537 • Successful completion of a CloseEnumeration operation
- 1538 • Successful completion of an Open or Pull operation that has its *EndOfSequence* operation
1539 output parameter set to TRUE. In other words, reaching the end of the enumeration set closes
1540 the enumeration session implicitly
- 1541 • Unsuccessful completion of a Pull operation when *ContinueOnError* had not been requested
- 1542 • WBEM server side decision to close the enumeration session based upon an operation timeout
- 1543 • WBEM server side decision to close an enumeration session during an operation on that
1544 enumeration session based upon exceeding server limits.

1545 A conformant WBEM server may support closure of enumeration sessions based upon exceeding server
 1546 limits. Potential examples for such a decision may be Pull operations with no objects requested that are
 1547 repeated with a high frequency on the same enumeration session, or EnumerationCount operations
 1548 repeated with a high frequency on the same enumeration session. If a WBEM server supports closure of
 1549 enumeration sessions based upon exceeding server limits, it shall make the decision to close an
 1550 enumeration session during an operation on that enumeration session. (There is no way to indicate the
 1551 reason for the closure if the decision is made elsewhere.)

1552 **6.5.2 Common operation parameters for the open operations**

1553 This subclause defines commonly used operation parameters for the Open operations. The description of
 1554 the individual Open operations references these operation parameters as appropriate. However, not
 1555 every Open operation uses every one of these common operation parameters.

1556 **6.5.2.1 EnumerationContext**

1557 The *EnumerationContext* operation output parameter is the enumeration context value representing the
 1558 enumeration session. See 6.5.1 for a definition of the concepts of *enumeration session* and *enumeration*
 1559 *context value*.

1560 **6.5.2.2 EndOfSequence**

1561 NOTE: This operation output parameter is also used for Pull operations.

1562 The *EndOfSequence* operation output parameter indicates whether the enumeration session is
 1563 exhausted.

1564 If *EndOfSequence* is TRUE upon successful completion of an operation, no more objects are available
 1565 and the WBEM server shall have closed the enumeration session, releasing any possibly allocated
 1566 resources related to the enumeration session.

1567 If the returned enumeration set is empty, it is valid for a WBEM server to set *EndOfSequence* to TRUE,
 1568 even if *MaxObjectCount* was 0. In this case, the enumeration session will be closed upon successful
 1569 completion of the operation.

1570 If *EndOfSequence* is FALSE upon successful completion of an operation, there may be additional
 1571 elements available and the WBEM server shall not have closed the enumeration session.

1572 **6.5.2.3 FilterQueryLanguage and FilterQueryString**

1573 The *FilterQueryLanguage* and *FilterQueryString* operation input parameters define a filter query that acts
 1574 as an additional restricting filter on the set of instances about which information is returned (that is, the
 1575 instances themselves or their instance paths).

1576 Support for the *FilterQueryLanguage* and *FilterQueryString* operation parameters is conditional on
 1577 support in the WBEM protocol for filter queries in pulled instance enumeration operations.

1578 If the WBEM protocol supports filter queries in pulled instance enumeration operations, the following rules
 1579 apply:

- 1580 • If *FilterQueryLanguage* is not NULL, additional filtering is requested and the following rules
 1581 apply:
 - 1582 – *FilterQueryLanguage* shall specify a valid query language and *FilterQueryString* shall
 1583 be a valid query in that query language. Neither the query language nor the format of
 1584 the filter query is defined by this specification. Conformant WBEM protocols shall
 1585 define a mechanism whereby WBEM servers can declare the set of query languages
 1586 that are valid for *FilterQueryLanguage*.

- 1587 – A filter query may specify any result set (e.g., SELECT list), but because the purpose
 1588 of the filter query is to restrict the set of instances about which information is returned,
 1589 its result set shall be ignored. The filter query shall not define any ordering criteria.
 1590 The filter query shall not define any grouping of objects. Operations using filter queries
 1591 may specify additional constraints on the filter query.
- 1592 – If the WBEM server infrastructure does not support filtered enumerations, the WBEM
 1593 server shall return failure with message WIPG0237 (Filter queries not supported by
 1594 WBEM server infrastructure).
- 1595 – If the CIM class implementation does not support filtered enumerations, the WBEM
 1596 server shall return failure with message WIPG0244 (Filter queries not supported by
 1597 class implementation).
- 1598 • If *FilterQueryLanguage* is NULL, no additional filtering shall take place, and *FilterQueryString*
 1599 shall be NULL.
 - 1600 – If *FilterQueryString* is not NULL, the WBEM server shall return failure with message
 1601 WIPG0208 (Invalid operation input parameter value).

1602 If the WBEM protocol does not support filter queries in pulled instance enumeration operations, no
 1603 additional filtering shall take place.

1604 6.5.2.4 OperationTimeout

1605 The *OperationTimeout* operation input parameter determines the "operation timeout". The operation
 1606 timeout is the minimum time the WBEM server shall maintain the open enumeration session after the last
 1607 Open or Pull operation (unless the enumeration session was closed during that last operation). If the
 1608 operation timeout is exceeded, the WBEM server may close the enumeration session at any time,
 1609 releasing any possibly allocated resources related to the enumeration session.

1610 Support for the *OperationTimeout* operation parameter in a conformant WBEM protocol is mandatory.

1611 An *OperationTimeout* of 0 means that there is no operation timeout, i.e., the enumeration session is never
 1612 closed based on time.

1613 If *OperationTimeout* is NULL, the WBEM server shall choose an operation timeout.

1614 All other values for *OperationTimeout* specify the operation timeout in seconds.

1615 A WBEM server may restrict the set of allowable values for *OperationTimeout*. This specifically includes
 1616 the possibility for the WBEM server to not allow 0 (no timeout). If the specified value is not an allowable
 1617 value, the WBEM server shall return failure with error message WIPG0242 (Invalid timeout). Conformant
 1618 WBEM protocols shall define a mechanism whereby WBEM servers can declare the allowable values for
 1619 *OperationTimeout*.

1620 6.5.2.5 ContinueOnError

1621 The *ContinueOnError* operation input parameter, if TRUE, requests continuation on error. Continuation on
 1622 error is the ability to resume an enumeration session successfully after a Pull operation that returned an
 1623 error. A conformant WBEM server may support continuation on error. Conformant WBEM protocols shall
 1624 define a mechanism whereby WBEM servers can declare support for continuation on error.

1625 Support for the *ContinueOnError* operation parameter is conditional on support in the WBEM protocol for
 1626 client side control of continuation on error for pulled instance enumeration operations.

1627 If the WBEM protocol supports client side control of continuation on error for pulled instance enumeration
 1628 operations, the following rules apply:

- 1629 • If a WBEM server does not support continuation on error and if *ContinueOnError* is TRUE, it
1630 shall return failure with error message WIPG0235 (Continuation on error not supported).
- 1631 • If a WBEM server supports continuation on error, it shall support it as follows: If
1632 *ContinueOnError* is TRUE, the enumeration session shall remain open when a Pull operation
1633 returns failure, and any subsequent successful Pull operations shall return the set of elements
1634 that would have been returned if the failing Pull operations had been successful, subject to the
1635 consistency rules defined in 5.8. If *ContinueOnError* is FALSE, the enumeration session shall
1636 be closed when a Pull operation returns failure.

1637 If the WBEM protocol does not support client side control of continuation on error for pulled instance
1638 enumeration operations, it shall define requirements for the behavior of the WBEM server with respect to
1639 continuation on error.

1640 **6.5.2.6 MaxObjectCount**

1641 NOTE: This operation output parameter is also used for Pull operations.

1642 The *MaxObjectCount* operation input parameter defines the maximum number of objects that may be
1643 returned by this operation. Any uint32 number is valid, including 0. The WBEM server may deliver any
1644 number of objects up to *MaxObjectCount* but shall not deliver more than *MaxObjectCount* objects.

1645 Support for the *MaxObjectCount* operation parameter in a conformant WBEM protocol is mandatory.

1646 A conformant WBEM server implementation may choose to never return any elements during an
1647 operation, regardless of the value of *MaxObjectCount*.

1648 A WBEM client may use a *MaxObjectCount* value of 0 to specify that it does not want to retrieve any
1649 instances in the operation.

1650 **6.5.3 OpenEnumerateInstances**

1651 **Purpose:**

1652 Establish and open an enumeration session for enumerating the instances of a class (including
1653 instances of its subclasses), and optionally retrieve a first set of instances.

1654 **Operation Input Parameters:**

1655

Generic Name	Generic Type	Requirement	Description
EnumClassPath	ClassPath	Mandatory	Class path of CIM class used for the enumeration (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 returned, 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.

Generic Name	Generic Type	Requirement	Description
IncludeClassOrigin	boolean	Conditional	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
ExcludeSubclass-Properties	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 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

1656
1657**Operation Output Parameters:**

Generic Name	Generic Type	Requirement	Description
InstanceList	InstanceSpecificationWithPath []	Mandatory	Sequence of instances with their instance paths of the first set of instances
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

1658

Description:

1659 The *OpenEnumerateInstances* operation establishes and opens an enumeration session for
 1660 enumerating all CIM instances of the class referenced by *EnumClassPath*, including instances of any
 1661 of its subclasses. Retrieval of a first set of those instances together with their instance paths may be
 1662 requested by setting *MaxObjectCount* to a value > 0.

1663 The set of instances to be returned throughout the entire enumeration session shall be determined
 1664 using the following algorithm:

- 1665 • Initially, the set of instances to be returned is the set of instances in the namespace
 1666 specified in *EnumClassPath*, whose creation class is the class specified in *EnumClassPath*
 1667 or a subclass of that class.
- 1668 • If the WBEM protocol supports filter queries for pulled instance enumeration operations
 1669 (that is, the *FilterQueryString* and *FilterQueryLanguage* operation parameters) and
 1670 *FilterQueryLanguage* is not NULL, *FilterQueryString* acts as a restricting filter on the

1671 instances to be returned such that any instances not selected by the filter query for its
 1672 result set are removed from the set of instances. The filter query shall query only the class
 1673 specified in *EnumClassPath*. See also 6.5.2.3.

1674 The set of instances to be returned throughout the entire enumeration session should not contain
 1675 any duplicate instances, as defined in 5.8.4. Because the set of returned instances contains only
 1676 instances that exist in the same namespace, a determination of duplicate instances (for example by
 1677 a WBEM client) can be done on the basis of their model paths only.

1678 The set of instances to be returned in the *InstanceList* operation parameter is the first set of
 1679 instances from the set of instances to be returned throughout the entire enumeration session, such
 1680 that no more than *MaxObjectCount* instances are returned. Returning no instances does not imply
 1681 that the enumeration session has been exhausted. Only the *EndOfSequence* operation output
 1682 parameter indicates whether the enumeration session has been exhausted.

1683 The set of properties to be included in any returned instances shall be determined using the following
 1684 algorithm:

- 1685 • Initially, the set of properties to be included is the set of properties exposed by the creation
 1686 class of the instance. This includes all the duplicates of any duplicate non-overridden
 1687 properties.
- 1688 • If the *IncludedProperties* operation input parameter is supported by the WBEM protocol
 1689 and if its value is not NULL, it acts as a restricting filter on the properties to be included in
 1690 the returned instances such that any properties exposed by the creation class of the
 1691 instance that are not named in that operation parameter are removed from the set of
 1692 properties to be included. Any duplicate or invalid property names in the
 1693 *IncludedProperties* operation input parameter shall be ignored. A non-NULL empty
 1694 *IncludedProperties* list removes all properties from the set of properties to be included.
- 1695 • If the *ExcludeSubclassProperties* operation input parameter is supported by the WBEM
 1696 protocol and if its value is TRUE, it acts as a restricting filter on the properties to be
 1697 included in the returned instances such that any properties not exposed by the class
 1698 referenced by *EnumClassPath* are removed from the set of properties to be included. In
 1699 other words, the set of properties is restricted to the properties exposed by the
 1700 enumeration class.
- 1701 • Conformant WBEM protocols may specify rules that cause properties with a value of NULL
 1702 to be removed from the set of properties to be included.

1703 **Preconditions:**

- 1704 • The CIM class referenced by *EnumClassPath* shall exist in the namespace. If this is not
 1705 satisfied, the operation shall fail, indicating WIPG0214.
- 1706 • If a filter query is specified,
 - 1707 – the query language specified in the *FilterQueryLanguage* operation parameter shall be
 1708 valid. If this is not satisfied, the operation shall fail, indicating WIPG0221.
 - 1709 – the query specified in the *FilterQueryString* operation parameter shall be a valid query in
 1710 the query language specified in the *FilterQueryLanguage* operation parameter. If this is not
 1711 satisfied, the operation shall fail, indicating WIPG0222 or WIPG0223.

1712 **Postconditions:**

- 1713 • The enumeration session shall have been established and opened.
- 1714 • A first set of instances with their instance paths shall have been returned as described in the
 1715 Description paragraph for this operation.

- 1716 • Requirements on ACID properties:
- 1717 – Atomicity: Required (related to the creation of an enumeration context that is maintained by
- 1718 the WBEM server)
- 1719 – Update Consistency: N/A
- 1720 – Isolation: Required at the level of single instances, as defined in 5.8.
- 1721 – Durability: Required (related to creation of an enumeration context that is maintained by
- 1722 the WBEM server)

1723 **Error Messages:**

1724

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0240	WBEM service limits are exceeded	Optional	Infrastructure, class implem.	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM service 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 service 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.	
WIPG0214	Class not found	Mandatory	Infrastructure	
WIPG0228	Operation not supported by class implementation	Mandatory	Class implem.	

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

1725 **6.5.4 OpenEnumerateInstancePaths**

1726 **Purpose:**

1727 Establish and open an enumeration session for enumerating the instances of a class (including
 1728 instances of its subclasses), and optionally retrieve a first set of instance paths of those instances.

1729 **Operation Input Parameters:**

1730

Generic Name	Generic Type	Requirement	Description
EnumClassPath	ClassPath	Mandatory	Class path of CIM class used for the enumeration (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 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

1731 **Operation Output Parameters:**

1732

Generic Name	Generic Type	Requirement	Description
InstancePathList	InstancePath []	Mandatory	Sequence of instance paths of the first set of instances

Generic Name	Generic Type	Requirement	Description
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

1733 **Description:**

1734 The *OpenEnumerateInstancePaths* operation establishes and opens an enumeration session for
 1735 enumerating the CIM instance paths of all instances of the class referenced by *EnumClassPath*,
 1736 including of instances of any of its subclasses. Retrieval of a first set of those instance paths may be
 1737 requested by setting *MaxObjectCount* to a value > 0.

1738 The set of instances from which instance paths are to be returned throughout the entire enumeration
 1739 session shall be determined using the following algorithm:

- 1740 • Initially, the set of instances to be returned is the set of instances in the namespace
 1741 specified in *EnumClassPath*, whose creation class is the class specified in *EnumClassPath*
 1742 or a subclass of that class.
- 1743 • If the WBEM protocol supports filter queries for pulled instance enumeration operations
 1744 (that is, the *FilterQueryString* and *FilterQueryLanguage* operation parameters) and
 1745 *FilterQueryLanguage* is not NULL, *FilterQueryString* acts as a restricting filter on the
 1746 instances to be returned such that any instances not selected by the filter query for its
 1747 result set are removed from the set of instances. The filter query shall query only the class
 1748 specified in *EnumClassPath*. See also 6.5.2.3.

1749 The set of instance paths to be returned throughout the entire enumeration session should not
 1750 contain any duplicate instance paths, as defined in 5.8.4. Because the instances referenced by the
 1751 set of returned instance paths contains only instances that exist in the same namespace, a
 1752 determination of duplicate instance paths can be done on the basis of their model paths only.

1753 The set of instance paths to be returned in the *InstancePathList* operation parameter is the first set of
 1754 instance paths from the set of instance paths to be returned throughout the entire enumeration
 1755 session, such that no more than *MaxObjectCount* instance paths are returned. Returning no instance
 1756 paths does not imply that the enumeration session has been exhausted. Only the *EndOfSequence*
 1757 operation output parameter indicates whether the enumeration session has been exhausted.

1758 **Preconditions:**

- 1759 • The CIM class referenced by *EnumClassPath* shall exist in the namespace. If this is not
 1760 satisfied, the operation shall fail, indicating WIPG0214.
- 1761 • If a filter query is specified,
 - 1762 – the query language specified in the *FilterQueryLanguage* operation parameter shall be
 1763 valid. If this is not satisfied, the operation shall fail, indicating WIPG0221.
 - 1764 – the query specified in the *FilterQueryString* operation parameter shall be a valid query in
 1765 the query language specified in the *FilterQueryLanguage* operation parameter. If this is not
 1766 satisfied, the operation shall fail, indicating WIPG0222 or WIPG0223.

1767 **Postconditions:**

- 1768 • The enumeration session shall have been established and opened.
- 1769 • A first set of instance paths shall have been returned as described in the Description paragraph
 1770 for this operation.

- 1771 • Requirements on ACID properties:
 - 1772 – Atomicity: Required (related to the creation of an enumeration context that is maintained by
 - 1773 the WBEM server)
 - 1774 – Update Consistency: N/A
 - 1775 – Isolation: Required at the level of single instances, as defined in 5.8.
 - 1776 – Durability: Required (related to creation of an enumeration context that is maintained by
 - 1777 the WBEM server)

1778 **Error Messages:**

1779

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0240	WBEM service limits are exceeded	Optional	Infrastructure, class implem.	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM service 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 service 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.	
WIPG0214	Class not found	Mandatory	Infrastructure	
WIPG0228	Operation not supported by class implementation	Mandatory	Class implem.	

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

1780 **6.5.5 OpenAssociators**1781 **Purpose:**

1782 Establish and open an enumeration session for enumerating instances that are associated with a
1783 given source instance, and optionally retrieve a first set of those instances.

1784 **Operation Input Parameters:**

1785

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

Generic Name	Generic Type	Requirement	Description
ExcludeSubclass-Properties	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

1786 **Operation Output Parameters:**
1787

Generic Name	Generic Type	Requirement	Description
InstanceList	InstanceSpecificationWithPath []	Mandatory	Sequence of instances with their instance paths of the first set of instances
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

1788 **Description:**

1789 The *OpenAssociators* operation establishes and opens an enumeration session for enumerating
1790 instances that are associated with the specified source instance. Retrieval of a first set of those
1791 instances together with their instance paths may be requested by setting *MaxObjectCount* to a value
1792 > 0.

1793 The set of instances to be returned throughout the entire enumeration session shall be determined
1794 using the following algorithm:

- 1795 • Initially, the set of instances to be returned is the set of all instances associated to the
1796 source instance specified in *SourceInstancePath*. These associations may be instances of
1797 different association classes.

1798 The result set should not contain any duplicate instances, as defined in 5.8.4. However,
1799 different far ends may reference the same instance, and in such cases, the instance shall
1800 be contained in the result set once for each such reference.

- 1801 • If the *AssociationClassName* operation input parameter is not NULL, it acts as a restricting
1802 filter on the instances to be returned such that each instance that is associated with the
1803 source instance using an association whose creation class or one of its superclasses does
1804 not have the name specified in *AssociationClassName*, is removed from the set of
1805 instances to be returned. There shall be no validity checking performed for the
1806 *AssociationClassName* operation input parameter.

- 1807 • If the *AssociatedClassName* operation input parameter is not NULL, it acts as a restricting
1808 filter on the instances to be returned such that each instance whose creation class or one
1809 of its superclasses does not have the name specified in *AssociatedClassName*, is removed
1810 from the set of instances to be returned. There shall be no validity checking performed for
1811 the *AssociatedClassName* operation input parameter.
- 1812 • If the *SourceRoleName* operation input parameter is not NULL, it acts as a restricting filter
1813 on the instances to be returned such that each instance that is associated with the source
1814 instance using an association class that has a role name on the source end that is not the
1815 role name specified in *SourceRoleName*, is removed from the set of instances to be
1816 returned. There shall be no validity checking performed for the *SourceRoleName* operation
1817 input parameter.
- 1818 • If the *AssociatedRoleName* operation input parameter is not NULL, it acts as a restricting
1819 filter on the instances to be returned such that each instance that is associated with the
1820 source instance using an association class that has a role name on the end referencing
1821 that instance that is not the role name specified in *AssociatedRoleName*, is removed from
1822 the set of instances to be returned. There shall be no validity checking performed for the
1823 *AssociatedRoleName* operation input parameter.
- 1824 • If the WBEM protocol supports filter queries for pulled instance enumeration operations
1825 (that is, the *FilterQueryString* and *FilterQueryLanguage* operation parameters) and
1826 *FilterQueryLanguage* is not NULL, *FilterQueryString* acts as a restricting filter on the
1827 instances to be returned such that any instances not selected by the filter query for its
1828 result set are removed from the set of instances. The filter query shall query only the class
1829 specified in *AssociatedClassName* (e.g., in the CQL FROM-clause). See also 6.5.2.3.
- 1830 The set of instances to be returned throughout the entire enumeration session should not contain
1831 any duplicate instances, as defined in 5.8.4. Because the set of returned instances contains only
1832 instances that exist in the same namespace, a determination of duplicate instances can be done on
1833 the basis of their model paths only.
- 1834 The set of instances to be returned in the *InstanceList* operation parameter is the first set of
1835 instances from the set of instances to be returned throughout the entire enumeration session, such
1836 that no more than *MaxObjectCount* instances are returned. Returning no instances does not imply
1837 that the enumeration session has been exhausted. Only the *EndOfSequence* operation output
1838 parameter indicates whether the enumeration session has been exhausted.
- 1839 The set of properties to be included in any returned instances shall be determined using the following
1840 algorithm:
- 1841 • Initially, the set of properties to be included is the set of properties exposed by the creation
1842 class of the instance. This includes all the duplicates of any duplicate non-overridden
1843 properties.
- 1844 • If the *IncludedProperties* operation input parameter is supported by the WBEM protocol
1845 and if its value is not NULL, it acts as a restricting filter on the properties to be included in
1846 the returned instances such that any properties exposed by the creation class of the
1847 instance that are not named in that operation parameter are removed from the set of
1848 properties to be included. Any duplicate or invalid property names in the
1849 *IncludedProperties* operation input parameter shall be ignored. A non-NULL empty
1850 *IncludedProperties* list removes all properties from the set of properties to be included.
- 1851 • If the *ExcludeSubclassProperties* operation input parameter is supported by the WBEM
1852 protocol and if its value is TRUE, it acts as a restricting filter on the properties to be
1853 included in the returned instances such that any properties not exposed by the class
1854 specified in *AssociatedClassName* are removed from the set of properties to be included.

- 1855 • Conformant WBEM protocols may specify rules that cause properties with a value of NULL
 1856 to be removed from the set of properties to be included.

1857 **Preconditions:**

- 1858 • The instance referenced by *SourceInstancePath* shall exist in the namespace. If this is not
 1859 satisfied, the operation shall fail, indicating WIPG0213.
- 1860 • If a filter query is specified,
- 1861 – the query language specified in the *FilterQueryLanguage* operation parameter shall be
 1862 valid. If this is not satisfied, the operation shall fail, indicating WIPG0221.
- 1863 – the query specified in the *FilterQueryString* operation parameter shall be a valid query in
 1864 the query language specified in the *FilterQueryLanguage* operation parameter. If this is not
 1865 satisfied, the operation shall fail, indicating WIPG0222 or WIPG0223.
- 1866 – the *AssociatedClassName* operation input parameter shall be non-NULL. If this is not
 1867 satisfied, the operation shall fail, indicating WIPG0208.
- 1868 • The *IncludedProperties* operation parameter, if supported by the WBEM protocol, shall only be
 1869 specified with a non-NULL value if the *AssociatedClassName* operation input parameter is also
 1870 non-NULL. If this is not satisfied, the operation shall fail, indicating WIPG0208.
- 1871 • The *ExcludeSubclassProperties* operation parameter, if supported by the WBEM protocol, shall
 1872 only be specified with a TRUE value if the *AssociatedClassName* operation input parameter is
 1873 non-NULL. If this is not satisfied, the operation shall fail, indicating WIPG0208.

1874 NOTE: Specifying a non-NULL value for *AssociatedClassName* ensures that the associated instances have the
 1875 class specified in *AssociatedClassName* as a common superclass.

1876 **Postconditions:**

- 1877 • The enumeration session shall have been established and opened.
- 1878 • A first set of instances with their instance paths shall have been returned as described in the
 1879 Description paragraph for this operation.
- 1880 • Requirements on ACID properties:
- 1881 – Atomicity: Required (related to the creation of an enumeration context that is maintained by
 1882 the WBEM server)
- 1883 – Update Consistency: N/A
- 1884 – Isolation: Required at the level of single instances, as defined in 5.8.
- 1885 – Durability: Required (related to creation of an enumeration context that is maintained by
 1886 the WBEM server)

1887 **Error Messages:**

1888

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0240	WBEM service limits are exceeded	Optional	Infrastructure, class implem.	
WIPG0204	Namespace not found	Mandatory	Infrastructure	

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0203	Operation not supported by WBEM service 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 service 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.	
WIPG0213	Instance not found	Mandatory	Class implem.	
WIPG0243	Timeout	Optional	Infrastructure, class implem.	
WIPG0227	Other failure	Optional	Infrastructure, class implem.	

1889 6.5.6 OpenAssociatorPaths

1890 Purpose:

1891 Establish and open an enumeration session for enumerating the instance paths of instances that are
1892 associated with a given source instance, and optionally retrieve a first set of those instance paths.

1893 Operation Input Parameters:

1894

Generic Name	Generic Type	Requirement	Description
SourceInstancePath	InstancePath	Mandatory	Instance path of the source instance (Context Parameter)

Generic Name	Generic Type	Requirement	Description
AssociationClass-Name	ClassName	Mandatory	NULL, or name of the association class, acting as a restricting filter on the returned instance paths
AssociatedClass-Name	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

1895
1896

Operation Output Parameters:

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

1897 **Description:**

1898 The *OpenAssociatorPaths* operation establishes and opens an enumeration session for enumerating
 1899 the instance paths of instances that are associated with the specified source instance. Retrieval of a
 1900 first set of those instance paths may be requested by setting *MaxObjectCount* to a value > 0.

1901 The set of instances of which instance paths are to be returned throughout the entire enumeration
 1902 session shall be determined using the following algorithm:

- 1903 • Initially, the set of instances to be returned is the set of all instances associated to the
 1904 source instance specified in *SourceInstancePath*. These associations may be instances of
 1905 different association classes.

1906 The result set should not contain any duplicate instance paths, as defined in 5.8.4.
 1907 However, different far ends may reference the same instance, and in such cases, the
 1908 instance path shall be contained in the result set once for each such reference.

- 1909 • If the *AssociationClassName* operation input parameter is not NULL, it acts as a restricting
 1910 filter on the instances to be returned such that each instance that is associated with the
 1911 source instance using an association whose creation class or one of its superclasses does
 1912 not have the name specified in *AssociationClassName*, is removed from the set of
 1913 instances to be returned. There shall be no validity checking performed for the
 1914 *AssociationClassName* operation input parameter.

- 1915 • If the *AssociatedClassName* operation input parameter is not NULL, it acts as a restricting
 1916 filter on the instances to be returned such that each instance whose creation class or one
 1917 of its superclasses does not have the name specified in *AssociatedClassName*, is removed
 1918 from the set of instances to be returned. There shall be no validity checking performed for
 1919 the *AssociatedClassName* operation input parameter.

- 1920 • If the *SourceRoleName* operation input parameter is not NULL, it acts as a restricting filter
 1921 on the instances to be returned such that each instance that is associated with the source
 1922 instance using an association class that has a role name on the source end that is not the
 1923 role name specified in *SourceRoleName*, is removed from the set of instances to be
 1924 returned. There shall be no validity checking performed for the *SourceRoleName* operation
 1925 input parameter.

- 1926 • If the *AssociatedRoleName* operation input parameter is not NULL, it acts as a restricting
 1927 filter on the instances to be returned such that each instance that is associated with the
 1928 source instance using an association class that has a role name on the end referencing
 1929 that instance that is not the role name specified in *AssociatedRoleName*, is removed from
 1930 the set of instances to be returned. There shall be no validity checking performed for the
 1931 *AssociatedRoleName* operation input parameter.

- 1932 • If the WBEM protocol supports filter queries for pulled instance enumeration operations
 1933 (that is, the *FilterQueryString* and *FilterQueryLanguage* operation parameters) and
 1934 *FilterQueryLanguage* is not NULL, *FilterQueryString* acts as a restricting filter on the
 1935 instances to be returned such that any instances not selected by the filter query for its
 1936 result set are removed from the set of instances. The filter query shall query only the class
 1937 specified in *AssociatedClassName* (e.g., in the CQL FROM-clause). See also 6.5.2.3.

1938 The set of instance paths to be returned throughout the entire enumeration session should not
 1939 contain any duplicate instance paths, as defined in 5.8.4. Because the set of returned instance paths
 1940 references only instances in the same namespace, a determination of duplicate instance paths can
 1941 be done on the basis of their model paths only.

1942 The set of instance paths to be returned in the *InstancePathList* operation parameter is the first set of
 1943 instance paths from the set of instance paths to be returned throughout the entire enumeration
 1944 session, such that no more than *MaxObjectCount* instance paths are returned. Returning no instance

1945 paths does not imply that the enumeration session has been exhausted. Only the *EndOfSequence*
 1946 operation output parameter indicates whether the enumeration session has been exhausted.

1947 **Preconditions:**

- 1948 • The instance referenced by *SourceInstancePath* shall exist in the namespace. If this is not
 1949 satisfied, the operation shall fail, indicating WIPG0213.
- 1950 • If a filter query is specified,
 - 1951 – the query language specified in the *FilterQueryLanguage* operation parameter shall be
 1952 valid. If this is not satisfied, the operation shall fail, indicating WIPG0221.
 - 1953 – the query specified in the *FilterQueryString* operation parameter shall be a valid query in
 1954 the query language specified in the *FilterQueryLanguage* operation parameter. If this is not
 1955 satisfied, the operation shall fail, indicating WIPG0222 or WIPG0223.
 - 1956 – the *AssociatedClassName* operation input parameter shall be non-NULL. If this is not
 1957 satisfied, the operation shall fail, indicating WIPG0208.

1958 NOTE: Specifying a non-NULL value for *AssociatedClassName* ensures that the associated instances have the
 1959 class specified in *AssociatedClassName* as a common superclass.

1960 **Postconditions:**

- 1961 • The enumeration session shall have been established and opened.
- 1962 • A first set of instance paths shall have been returned as described in the Description paragraph
 1963 for this operation.
- 1964 • Requirements on ACID properties:
 - 1965 – Atomicity: Required (related to the creation of an enumeration context that is maintained by
 1966 the WBEM server)
 - 1967 – Update Consistency: N/A
 - 1968 – Isolation: Required at the level of single instances, as defined in 5.8.
 - 1969 – Durability: Required (related to creation of an enumeration context that is maintained by
 1970 the WBEM server)

1971 **Error Messages:**

1972

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0240	WBEM service limits are exceeded	Optional	Infrastructure, class implem.	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM service 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	
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 service 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.	
WIPG0213	Instance not found	Mandatory	Class implem.	
WIPG0243	Timeout	Optional	Infrastructure, class implem.	
WIPG0227	Other failure	Optional	Infrastructure, class implem.	

1973 6.5.7 OpenReferences

1974 Purpose:

1975 Establish and open an enumeration session for enumerating the association instances that reference
 1976 a given source instance, and optionally retrieve a first set of those instances.

1977 Operation Input Parameters:

1978

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

Generic Name	Generic Type	Requirement	Description
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
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	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
ExcludeSubclass-Properties	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

1979
1980

Operation Output Parameters:

Generic Name	Generic Type	Requirement	Description
InstanceList	InstanceSpecificationWithPath []	Mandatory	Sequence of instances with their instance paths of the first set of instances
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

1981 **Description:**

1982 The *OpenReferences* operation establishes and opens an enumeration session for enumerating the
 1983 association instances that reference the specified source instance. Retrieval of a first set of those
 1984 instances together with their instance paths may be requested by setting *MaxObjectCount* to a value
 1985 > 0.

1986 The set of instances to be returned throughout the entire enumeration session shall be determined
 1987 using the following algorithm:

- 1988 • Initially, the set of instances to be returned is the set of all instances referencing the source
 1989 instance specified in *SourceInstancePath*. These associations may be instances of
 1990 different association classes.
- 1991 • If the *AssociationClassName* operation input parameter is not NULL, it acts as a restricting
 1992 filter on the instances to be returned such that each association instance whose creation
 1993 class or one of its superclasses does not have the name specified in
 1994 *AssociationClassName*, is removed from the set of instances to be returned. There shall be
 1995 no validity checking performed for the *AssociationClassName* operation input parameter.
- 1996 • If the *AssociatedClassName* operation input parameter is not NULL, it acts as a restricting
 1997 filter on the instances to be returned such that each association instance whose creation
 1998 class has a set of far ends none of which is referencing a class where that class or one of
 1999 its superclasses has the name specified in *AssociatedClassName*, is removed from the set
 2000 of instances to be returned. There shall be no validity checking performed for the
 2001 *AssociatedClassName* operation input parameter.
- 2002 • If the *SourceRoleName* operation input parameter is not NULL, it acts as a restricting filter
 2003 on the instances to be returned such that each association instance whose creation class
 2004 does not have the role name specified in *SourceRoleName* on the end referencing the
 2005 source instance, is removed from the set of instances to be returned. There shall be no
 2006 validity checking performed for the *SourceRoleName* operation input parameter.
- 2007 • If the *AssociatedRoleName* operation input parameter is not NULL, it acts as a restricting
 2008 filter on the instances to be returned such that each association instance whose creation
 2009 class has a set of far ends none of which has the role name specified in
 2010 *AssociatedRoleName*, is removed from the set of instances to be returned. There shall be
 2011 no validity checking performed for the *AssociatedRoleName* operation input parameter.
- 2012 • If the WBEM protocol supports filter queries for pulled instance enumeration operations
 2013 (that is, the *FilterQueryString* and *FilterQueryLanguage* operation parameters) and
 2014 *FilterQueryLanguage* is not NULL, *FilterQueryString* acts as a restricting filter on the
 2015 instances to be returned such that any instances not selected by the filter query for its
 2016 result set are removed from the set of instances. The filter query shall query only the class
 2017 specified in *AssociationClassName* (e.g., in the CQL FROM-clause). See also 6.5.2.3.

2018 The set of instances to be returned throughout the entire enumeration session should not contain
 2019 any duplicate instances, as defined in 5.8.4. Because the set of returned instances contains only
 2020 instances that exist in the same namespace, so any determination of duplicate instances (for
 2021 example by a WBEM client) may be done on the basis of their model paths.

2022 The set of instances to be returned in the *InstanceList* operation parameter is the first set of
 2023 instances from the set of instances to be returned throughout the entire enumeration session, such

2024 that no more than *MaxObjectCount* instances are returned. Returning no instances does not imply
 2025 that the enumeration session has been exhausted. Only the *EndOfSequence* operation output
 2026 parameter indicates whether the enumeration session has been exhausted.

2027 The set of properties to be included in any returned instances shall be determined using the following
 2028 algorithm:

- 2029 • Initially, the set of properties to be included is the set of properties exposed by the creation
 2030 class of the instance. This includes all the duplicates of any duplicate non-overridden
 2031 properties.
- 2032 • If the *IncludedProperties* operation input parameter is supported by the WBEM protocol
 2033 and if its value is not NULL, it acts as a restricting filter on the properties to be included in
 2034 the returned instances such that any properties exposed by the creation class of the
 2035 instance that are not named in that operation parameter are removed from the set of
 2036 properties to be included. Any duplicate or invalid property names in the
 2037 *IncludedProperties* operation input parameter shall be ignored. A non-NULL empty
 2038 *IncludedProperties* list removes all properties from the set of properties to be included.
- 2039 • If the *ExcludeSubclassProperties* operation input parameter is supported by the WBEM
 2040 protocol and if its value is TRUE, it acts as a restricting filter on the properties to be
 2041 included in the returned instances such that any properties not exposed by the class
 2042 specified in *AssociationClassName* are removed from the set of properties to be included.
- 2043 • Conformant WBEM protocols may specify rules that cause properties with a value of NULL
 2044 to be removed from the set of properties to be included.

2045 **Preconditions:**

- 2046 • The instance referenced by *SourceInstancePath* shall exist in the namespace. If this is not
 2047 satisfied, the operation shall fail, indicating WIPG0213.
- 2048 • If a filter query is specified,
 - 2049 – the query language specified in the *FilterQueryLanguage* operation parameter shall be
 2050 valid. If this is not satisfied, the operation shall fail, indicating WIPG0221.
 - 2051 – the query specified in the *FilterQueryString* operation parameter shall be a valid query in
 2052 the query language specified in the *FilterQueryLanguage* operation parameter. If this is not
 2053 satisfied, the operation shall fail, indicating WIPG0222 or WIPG0223.
 - 2054 – the *AssociationClassName* operation input parameter shall be non-NULL. If this is not
 2055 satisfied, the operation shall fail, indicating WIPG0208.
- 2056 • The *IncludedProperties* operation parameter, if supported by the WBEM protocol, shall only be
 2057 specified with a non-NULL value if the *AssociationClassName* operation input parameter is also
 2058 non-NULL. If this is not satisfied, the operation shall fail, indicating WIPG0208.
- 2059 • The *ExcludeSubclassProperties* operation parameter, if supported by the WBEM protocol, shall
 2060 only be specified with a TRUE value if the *AssociationClassName* operation input parameter is
 2061 non-NULL. If this is not satisfied, the operation shall fail, indicating WIPG0208.

2062 NOTE: Specifying a non-NULL value for *AssociationClassName* ensures that the association instances have the
 2063 class specified in *AssociationClassName* as a common superclass.

2064 **Postconditions:**

- 2065 • The enumeration session shall have been established and opened.
- 2066 • A first set of instances with their instance paths shall have been returned as described in the
 2067 Description paragraph for this operation.
- 2068 • Requirements on ACID properties:

- 2069 – Atomicity: Required (related to the creation of an enumeration context that is maintained by
2070 the WBEM server)
- 2071 – Update Consistency: N/A
- 2072 – Isolation: Required at the level of single instances, as defined in 5.8.
- 2073 – Durability: Required (related to creation of an enumeration context that is maintained by
2074 the WBEM server)

2075 **Error Messages:**
2076

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0240	WBEM service limits are exceeded	Optional	Infrastructure, class implem.	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM service 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 service 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.	
WIPG0213	Instance not found	Mandatory	Class implem.	
WIPG0243	Timeout	Optional	Infrastructure, class implem.	

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

2077 **6.5.8 OpenReferencePaths**

2078 **Purpose:**

2079 Establish and open an enumeration session for enumerating the instance paths of association
 2080 instances that reference a given source instance, and optionally retrieve a first set of those instance
 2081 paths.

2082 **Operation Input Parameters:**

2083

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

Generic Name	Generic Type	Requirement	Description
MaxObjectCount	uint32	Mandatory	Maximum number of instance paths that may be returned by this operation, as defined in 6.5.2.6

2084
2085**Operation Output Parameters:**

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

2086 **Description:**

2087 The *OpenReferencePaths* operation establishes and opens an enumeration session for enumerating
 2088 the instance paths of association instances that reference the specified source instance. Retrieval of
 2089 a first set of those instance paths may be requested by setting *MaxObjectCount* to a value > 0.

2090 The set of instances of which instance paths are to be returned throughout the entire enumeration
 2091 session shall be determined using the following algorithm:

- 2092 • Initially, the set of instances to be returned is the set of all instances referencing the source
 2093 instance specified in *SourceInstancePath*. These associations may be instances of
 2094 different association classes.
- 2095 • If the *AssociationClassName* operation input parameter is not NULL, it acts as a restricting
 2096 filter on the instances to be returned such that each association instance whose creation
 2097 class or one of its superclasses does not have the name specified in
 2098 *AssociationClassName*, is removed from the set of instances to be returned. There shall be
 2099 no validity checking performed for the *AssociationClassName* operation input parameter.
- 2100 • If the *AssociatedClassName* operation input parameter is not NULL, it acts as a restricting
 2101 filter on the instances to be returned such that each association instance whose creation
 2102 class has a set of far ends none of which is referencing a class where that class or one of
 2103 its superclasses has the name specified in *AssociatedClassName*, is removed from the set
 2104 of instances to be returned. There shall be no validity checking performed for the
 2105 *AssociatedClassName* operation input parameter.
- 2106 • If the *SourceRoleName* operation input parameter is not NULL, it acts as a restricting filter
 2107 on the instances to be returned such that each association instance whose creation class
 2108 does not have the role name specified in *SourceRoleName* on the end referencing the
 2109 source instance, is removed from the set of instances to be returned. There shall be no
 2110 validity checking performed for the *SourceRoleName* operation input parameter.
- 2111 • If the *AssociatedRoleName* operation input parameter is not NULL, it acts as a restricting
 2112 filter on the instances to be returned such that each association instance whose creation
 2113 class has a set of far ends none of which has the role name specified in
 2114 *AssociatedRoleName*, is removed from the set of instances to be returned. There shall be
 2115 no validity checking performed for the *AssociatedRoleName* operation input parameter.
- 2116 • If the WBEM protocol supports filter queries for pulled instance enumeration operations
 2117 (that is, the *FilterQueryString* and *FilterQueryLanguage* operation parameters) and

2118 *FilterQueryLanguage* is not NULL, *FilterQueryString* acts as a restricting filter on the
 2119 instances to be returned such that any instances not selected by the filter query for its
 2120 result set are removed from the set of instances. The filter query shall query only the class
 2121 specified in *AssociationClassName* (e.g., in the CQL FROM-clause). See also 6.5.2.3.

2122 The set of instance paths to be returned throughout the entire enumeration session should not
 2123 contain any duplicate instance paths, as defined in 5.8.4. Because the set of returned instance paths
 2124 references only instances that exist in the same namespace, a determination of duplicate instance
 2125 paths can be done on the basis of their model paths only.

2126 The set of instance paths to be returned in the *InstancePathList* operation parameter is the first set of
 2127 instance paths from the set of instance paths to be returned throughout the entire enumeration
 2128 session, such that no more than *MaxObjectCount* instances are returned. Returning no instance
 2129 paths does not imply that the enumeration session has been exhausted. Only the *EndOfSequence*
 2130 operation output parameter indicates whether the enumeration session has been exhausted.

2131 **Preconditions:**

- 2132 • The instance referenced by *SourceInstancePath* shall exist in the namespace. If this is not
 2133 satisfied, the operation shall fail, indicating WIPG0213.
- 2134 • If a filter query is specified,
 - 2135 – the query language specified in the *FilterQueryLanguage* operation parameter shall be
 2136 valid. If this is not satisfied, the operation shall fail, indicating WIPG0221.
 - 2137 – the query specified in the *FilterQueryString* operation parameter shall be a valid query in
 2138 the query language specified in the *FilterQueryLanguage* operation parameter. If this is not
 2139 satisfied, the operation shall fail, indicating WIPG0222 or WIPG0223.
 - 2140 – the *AssociationClassName* operation input parameter shall be non-NULL. If this is not
 2141 satisfied, the operation shall fail, indicating WIPG0208.

2142 NOTE: Specifying a non-NULL value for *AssociationClassName* ensures that the association instances have the
 2143 class specified in *AssociationClassName* as a common superclass.

2144 **Postconditions:**

- 2145 • The enumeration session shall have been established and opened.
- 2146 • A first set of instance paths shall have been returned as described in the Description paragraph
 2147 for this operation.
- 2148 • Requirements on ACID properties:
 - 2149 – Atomicity: Required (related to the creation of an enumeration context that is maintained by
 2150 the WBEM server)
 - 2151 – Update Consistency: N/A
 - 2152 – Isolation: Required at the level of single instances, as defined in 5.8.
 - 2153 – Durability: Required (related to creation of an enumeration context that is maintained by
 2154 the WBEM server)

2155 **Error Messages:**

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

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0240	WBEM service limits are exceeded	Optional	Infrastructure, class implem.	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM service 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 service 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.	
WIPG0213	Instance not found	Mandatory	Class implem.	
WIPG0243	Timeout	Optional	Infrastructure, class implem.	
WIPG0227	Other failure	Optional	Infrastructure, class implem.	

2157 **6.5.9 OpenQueryInstances**

2158 **Purpose:**

2159 Establish and open an enumeration session for enumerating the instances of a query result, and
 2160 optionally retrieve a first set of instances.

2161 **Operation Input Parameters:**
2162

Generic Name	Generic Type	Requirement	Description
NamespacePath	NamespacePath	Mandatory	Namespace path of the CIM namespace the query is executed in (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>
ReturnQueryResult-Class	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

2163 **Operation Output Parameters:**
2164

Generic Name	Generic Type	Requirement	Description
InstanceList	InstanceSpecification []	Mandatory	Sequence of instances of the first set of instances
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

2165 **Description:**

2166 The *OpenQueryInstances* operation establishes and opens an enumeration session for enumerating
2167 the instances representing the result of the query specified in *QueryString* in the CIM namespace
2168 referenced by *NamespacePath*. Retrieval of a first set of those instances may be requested by
2169 setting *MaxObjectCount* to a value > 0.

2170 The set of instances to be returned in the *InstanceList* operation parameter is the first set of
2171 instances from the set of instances to be returned throughout the entire enumeration session, such
2172 that no more than *MaxObjectCount* instances are returned. Returning no instances in the
2173 *InstanceList* operation parameter does not imply that the enumeration session has been exhausted.
2174 Only the *EndOfSequence* operation output parameter indicates whether the enumeration session
2175 has been exhausted.

2176 The returned instances are only representations of instances that have no corresponding
2177 addressable instances residing in the WBEM server.

2178 If *QueryLanguage* is not NULL, it shall specify a valid query language and *QueryString* shall be a
2179 valid query in that query language. Neither the query language nor the format of the filter query is
2180 defined by this specification. Conformant WBEM protocols shall specify a mechanism for determining
2181 the set of query languages that are valid for *QueryLanguage*. The simplest way to do this is to list the
2182 set of valid query languages.

2183 The value of the *ReturnQueryResultClass* operation input parameter controls whether or not a class
2184 definition is returned in the *QueryResultClass* operation output parameter. If FALSE, then
2185 *QueryResultClass* shall be NULL. If TRUE, then the value of *QueryResultClass* shall be a class
2186 definition that defines the properties of each instance of the query result. The name of this class shall
2187 be CIM_QueryResult. This class is only a representation of a class that has no corresponding
2188 addressable class residing in the WBEM server.

2189 Preconditions:

- 2190 • The CIM namespace referenced by *NamespacePath* shall exist. If this is not satisfied, the
2191 operation shall fail, indicating WIPG0204.
- 2192 • The query language specified in the *QueryLanguage* operation parameter shall be a valid query
2193 language. If this is not satisfied, the operation shall fail, indicating WIPG0221.
- 2194 • The query specified in the *QueryString* operation parameter shall be a valid query in the query
2195 language specified in the *QueryLanguage* operation parameter. If this is not satisfied, the
2196 operation shall fail, indicating WIPG0222 or WIPG0223.

2197 Postconditions:

- 2198 • The enumeration session shall have been established and opened.
- 2199 • A first set of instances shall have been returned as described in the Description paragraph for
2200 this operation.
- 2201 • Requirements on ACID properties:
 - 2202 – Atomicity: Required (related to the creation of an enumeration context that is maintained by
2203 the WBEM server)
 - 2204 – Update Consistency: N/A
 - 2205 – Isolation: Required at the level of single instances, as defined in 5.8.
 - 2206 – Durability: Required (related to creation of an enumeration context that is maintained by
2207 the WBEM server)

2208 Error Messages:

2209

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

2210 6.5.10 Common operation parameters for the pull operations

2211 This subclause defines commonly used operation parameters for the Pull operations. The description of
 2212 the individual Pull operations references these operation parameters as appropriate. However, not every
 2213 Pull operation uses every one of these common operation parameters.

2214 6.5.10.1 NamespacePath

2215 The *NamespacePath* operation input parameter references the CIM namespace identified by the context
 2216 parameter of the Open operation that established and opened the enumeration session.

2217 6.5.10.2 EnumerationContext

2218 The *EnumerationContext* operation input/output parameter is the enumeration context value representing
 2219 the enumeration session to be used.

2220 Support for the *EnumerationContext* operation parameter in a conformant WBEM protocol is mandatory.

2221 When invoking the Pull operation, the enumeration session represented by *EnumerationContext* shall be
 2222 open. The enumeration session shall have been established using one of the Open operations whose
 2223 type of enumerated element matches the Pull operation. For the first Pull operation on an enumeration
 2224 session, the value of *EnumerationContext* shall be the enumeration context value returned by a
 2225 successful Open operation that established and opened that enumeration session. For any subsequent
 2226 Pull operations on that enumeration session, the value of *EnumerationContext* shall be the value of
 2227 *EnumerationContext* as returned by the previous Pull operation on the same enumeration session.

2228 After completing the Pull operation, the enumeration session represented by *EnumerationContext* shall
2229 be open or closed.

2230 6.5.10.3 EndOfSequence

2231 The *EndOfSequence* operation output parameter when used in Pull operations behaves as defined in
2232 6.5.2.2

2233 6.5.10.4 MaxObjectCount

2234 The *MaxObjectCount* operation input parameter when used in Pull operations behaves as defined in
2235 6.5.2.6.

2236 6.5.11 PullInstancesWithPath

2237 Purpose:

2238 Retrieve the next set of instances together with their instance paths from an open enumeration
2239 session.

2240 Operation Input Parameters:

Generic Name	Generic Type	Requirement	Description
NamespacePath	NamespacePath	Mandatory	Namespace path of the CIM namespace, 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

2242 Operation Output Parameters:

Generic Name	Generic Type	Requirement	Description
InstanceList	InstanceSpecificationWithPath []	Mandatory	Sequence of instances with their instance paths of the retrieved set of instances
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

2244 Description:

2245 The *PullInstancesWithPath* operation retrieves the next set of instances together with their instance
2246 paths from an open enumeration session.

2247 The enumeration session shall have been established using one of the following operations:

- 2248 • OpenEnumerateInstances

2249 • OpenAssociators

2250 • OpenReferences

2251 The set of instances to be returned in the *InstanceList* operation parameter is the next set of
 2252 instances from the set of instances to be returned throughout the entire enumeration session, such
 2253 that no more than *MaxObjectCount* instances are returned. Returning no instances does not imply
 2254 that the enumeration session has been exhausted. Only the *EndOfSequence* operation output
 2255 parameter indicates whether the enumeration session has been exhausted.

2256 The set of properties to be included in any retrieved instances shall be the as determined using the
 2257 Open operation that established the enumeration session.

2258 **Preconditions:**

2259 • The enumeration session identified by *EnumerationContext* shall be open. If this is not satisfied,
 2260 the operation shall fail, indicating WIPG0241.

2261 • The value of *EnumerationContext* shall be the enumeration context value returned by the
 2262 previous Open or Pull operation on the same enumeration session. If this is not satisfied, the
 2263 operation shall fail, indicating WIPG0241.

2264 **Postconditions:**

2265 • The set of instances with their instance paths shall have been returned as described in the
 2266 Description paragraph for this operation.

2267 • Requirements on ACID properties:

2268 – Atomicity: Required (related to updates to an enumeration context that is maintained by the
 2269 WBEM server)

2270 – Update Consistency: N/A

2271 – Isolation: Required at the level of single instances, as defined in 5.8.

2272 – Durability: Required (related to updates to an enumeration context that is maintained by
 2273 the WBEM server)

2274 **Error Messages:**

2275

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0240	WBEM service limits are exceeded	Optional	Infrastructure, class implem.	
WIPG0203	Operation not supported by WBEM service 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
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.	

2276 6.5.12 PullInstancePaths

2277 Purpose:

2278 Retrieve the next set of instance paths from an open enumeration session.

2279 Operation Input Parameters:

2280

Generic Name	Generic Type	Requirement	Description
NamespacePath	NamespacePath	Mandatory	Namespace path of the CIM namespace, 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 instance paths that may be returned by this operation, as defined in 6.5.10.4

2281 Operation Output Parameters:

2282

Generic Name	Generic Type	Requirement	Description
InstancePathList	InstancePath []	Mandatory	Sequence of retrieved 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

2283 Description:

2284 The *PullInstancePaths* operation retrieves the next set of instance paths from an open enumeration session.

2286 The enumeration session shall have been established using one of the following operations:

- 2287 • OpenEnumerateInstancePaths
- 2288 • OpenAssociatorPaths
- 2289 • OpenReferencePaths

2290 The set of instance paths to be returned in the *InstancePathList* operation parameter is the next set
 2291 of instance paths from the set of instance paths to be returned throughout the entire enumeration
 2292 session, such that no more than *MaxObjectCount* instance paths are returned. Returning no instance
 2293 paths does not imply that the enumeration session has been exhausted. Only the *EndOfSequence*
 2294 operation output parameter indicates whether the enumeration session has been exhausted.

2295 **Preconditions:**

- 2296 • The enumeration session identified by *EnumerationContext* shall be open. If this is not satisfied,
 2297 the operation shall fail, indicating WIPG0241.
- 2298 • The value of *EnumerationContext* shall be the enumeration context value returned by the
 2299 previous Open or Pull operation on the same enumeration session. If this is not satisfied, the
 2300 operation shall fail, indicating WIPG0241.

2301 **Postconditions:**

- 2302 • The set of instance paths shall have been returned as described in the Description paragraph
 2303 for this operation.
- 2304 • Requirements on ACID properties:
 - 2305 – Atomicity: Required (related to updates to an enumeration context that is maintained by the
 2306 WBEM server)
 - 2307 – Update Consistency: N/A
 - 2308 – Isolation: Required at the level of single instances, as defined in 5.8.
 - 2309 – Durability: Required (related to updates to an enumeration context that is maintained by
 2310 the WBEM server)

2311 **Error Messages:**

2312

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0240	WBEM service limits are exceeded	Optional	Infrastructure, class implem.	
WIPG0203	Operation not supported by WBEM service 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.	
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.	

2313 6.5.13 PullInstances

2314 Purpose:

2315 Retrieve the next set of instances from an open enumeration session.

2316 Operation Input Parameters:

2317

Generic Name	Generic Type	Requirement	Description
NamespacePath	NamespacePath	Mandatory	Namespace path of the CIM namespace, 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

2318 Operation Output Parameters:

2319

Generic Name	Generic Type	Requirement	Description
InstanceList	InstanceSpecification []	Mandatory	Sequence of retrieved instances
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

2320 Description:

2321 The *PullInstancesWithPath* operation retrieves the next set of instances without their instance paths
2322 from an open enumeration session.

2323 The enumeration session shall have been established using one of the following operations:

- 2324 • *OpenQueryInstances*

2325 The set of instances to be returned in the *InstanceList* operation parameter is the next set of
 2326 instances from the set of instances to be returned throughout the entire enumeration session, such
 2327 that no more than *MaxObjectCount* instances are returned. Returning no instances does not imply
 2328 that the enumeration session has been exhausted. Only the *EndOfSequence* operation output
 2329 parameter indicates whether the enumeration session has been exhausted.

2330 The set of properties to be included in any retrieved instances shall be the as determined using the
 2331 Open operation that established the enumeration session.

2332 **Preconditions:**

- 2333 • The enumeration session identified by *EnumerationContext* shall be open. If this is not satisfied,
 2334 the operation shall fail, indicating WIPG0241.
- 2335 • The value of *EnumerationContext* shall be the enumeration context value returned by the
 2336 previous Open or Pull operation on the same enumeration session. If this is not satisfied, the
 2337 operation shall fail, indicating WIPG0241.

2338 **Postconditions:**

- 2339 • The set of instances shall have been returned as described in the Description paragraph for this
 2340 operation.
- 2341 • Requirements on ACID properties:
 - 2342 – Atomicity: Required (related to updates to an enumeration context that is maintained by the
 2343 WBEM server)
 - 2344 – Update Consistency: N/A
 - 2345 – Isolation: Required at the level of single instances, as defined in 5.8.
 - 2346 – Durability: Required (related to updates to an enumeration context that is maintained by
 2347 the WBEM server)

2348 **Error Messages:**

2349

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0240	WBEM service limits are exceeded	Optional	Infrastructure, class implem.	
WIPG0203	Operation not supported by WBEM service 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.	

Message ID	Message Name	Requirement	Sources	Additional Description
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.	

2350 6.5.14 CloseEnumeration

2351 Purpose:

2352 Close an open enumeration session.

2353 Operation Input Parameters:

2354

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

2355 Operation Output Parameters:

2356 None.

2357 Description:

2358 The *CloseEnumeration* operation closes the open enumeration session identified by
2359 *EnumerationContext*.

2360 The enumeration session shall have been established using any of the Open operations.

2361 Enumeration sessions are closed implicitly when exhausted, so this operation only needs to be used
2362 when terminating an enumeration sequence before it is exhausted.

2363 Preconditions:

- 2364 • The enumeration session identified by *EnumerationContext* shall be open. If this is not satisfied,
2365 the operation shall fail, indicating WIPG0241.
- 2366 • The value of *EnumerationContext* shall be the enumeration context value returned by the
2367 previous Open or Pull operation on the same enumeration session. If this is not satisfied, the
2368 operation shall fail, indicating WIPG0241.

2369 Postconditions:

- 2370 • The enumeration session identified by *EnumerationContext* is closed.
- 2371 • Requirements on ACID properties:

- 2372 – Atomicity: Required (related to updates to or deletion of an enumeration context that is
- 2373 – maintained by the WBEM server)
- 2374 – Update Consistency: N/A
- 2375 – Isolation: Required
- 2376 – Durability: Required (related to updates to or deletion of an enumeration context that is
- 2377 – maintained by the WBEM server)

2378 **Error Messages:**
2379

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0203	Operation not supported by WBEM service 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.	

2380 **6.5.15 EnumerationCount**

2381 **Purpose:**

2382 Estimate the total number of remaining items in an open enumeration session.

2383 **Operation Input Parameters:**
2384

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

2385 **Operation Output Parameters:**
2386

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

2387 **Description:**

2388 The *EnumerationCount* operation estimates the total number of remaining items in the open
2389 enumeration session identified by *EnumerationContext*.

2390 The enumeration session shall have been established using any of the Open operations.

2391 If not NULL, the *EnumerationCount* operation output parameter is an estimated count of the number
2392 of items remaining to be retrieved with subsequent Pull operations. Thus, executing this operation
2393 immediately after opening the enumeration session provides an estimate of the total number of items
2394 that will be returned in the enumeration set.

2395 If the WBEM server cannot or will not return an estimated count, it may respond with success and
2396 the NULL value in the *EnumerationCount* operation output parameter.

2397 This mechanism is intended to assist WBEM clients in determining the overall size of an
2398 enumeration set and of the number of items remaining in the enumeration session. However,
2399 because it is an estimate and not an exact number, it should not be used for determining the end of
2400 an enumeration sequence, i.e., in place of the *EndOfSequence* operation output parameter on Open
2401 and Pull operations.

2402 **Preconditions:**

- 2403 • The enumeration session identified by *EnumerationContext* shall be open. If this is not satisfied,
2404 the operation shall fail, indicating WIPG0241.
- 2405 • The value of *EnumerationContext* shall be the enumeration context value returned by the
2406 previous Open or Pull operation on the same enumeration session. If this is not satisfied, the
2407 operation shall fail, indicating WIPG0241.

2408 **Postconditions:**

- 2409 • Requirements on ACID properties:
 - 2410 – Atomicity: N/A
 - 2411 – Update Consistency: N/A
 - 2412 – Isolation: Required
 - 2413 – Durability: N/A

2414 **Error Messages:**
2415

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0240	WBEM service limits are exceeded	Optional	Infrastructure, class implem.	

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0203	Operation not supported by WBEM service 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.	

2416 **6.6 Method invocation operations**

2417 This subclause defines operations for the invocation of CIM methods.

2418 **6.6.1 InvokeMethod**

2419 **Purpose:**

2420 Invoke a CIM method using an instance path.

2421 **Operation Input Parameters:**

2422

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

2423 **Operation Output Parameters:**

2424

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

2425 **Description:**

2426 Invoke a CIM method using an instance path. The method may be static or non-static.

2427 Conformant WBEM protocols shall define a mapping for the invocation of CIM methods using an
 2428 instance path, including a mapping of the operation parameters defined in the tables above. These
 2429 rules may map the method invocation to a single operation, map each method to its own separate
 2430 operation, or define any other appropriate mapping.

2431 If the implementation of the method could be invoked, the operation is considered successful,
 2432 regardless of what the semantics of any return values or output parameters is. For example, if a
 2433 method defines that a particular return value indicates an error condition, the method invocation was
 2434 still successful from a perspective of the invocation operation.

2435 **Preconditions:**

- 2436 • The instance referenced by *InstancePath* shall exist in the namespace. If this is not satisfied,
 2437 the operation shall fail, indicating WIPG0213.
- 2438 • The method to be invoked shall be exposed by the creation class of the instance referenced by
 2439 *InstancePath*. If this is not satisfied, the operation shall fail, indicating WIPG0218.

2440 **Postconditions:**

- 2441 • The CIM method shall have been invoked.
- 2442 • Requirements on ACID properties:
 - 2443 – Atomicity: Recommended
 - 2444 – Update Consistency: Recommended
 - 2445 – Isolation: Recommended
 - 2446 – Durability: Required

2447 **Error Messages:**

2448

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0240	WBEM service limits are exceeded	Optional	Infrastructure, class implem.	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0229	Method invocation not supported by WBEM service 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	

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure, class implem.	
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.	

2449 **6.6.2 InvokeStaticMethod**

2450 **Purpose:**

2451 Invoke a static CIM method using a class path.

2452 **Operation Input Parameters:**

2453

Generic Name	Generic Type	Requirement	Description
ClassPath	ClassPath	Mandatory	Class path of the CIM 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

2454 **Operation Output Parameters:**

2455

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

2456 **Description:**

2457 Invoke a static CIM method using a class path.

2458 Conformant WBEM protocols shall define a mapping for the invocation of CIM methods using a class
 2459 path, including a mapping of the operation parameters defined in the tables above. These rules may
 2460 map the method invocation to a single operation, map each method to its own separate operation, or
 2461 define any other appropriate mapping.

2462 If the implementation of the method could be invoked, the operation is considered successful,
 2463 regardless of what the semantics of any return values or output parameters is. For example, if a
 2464 method defines that a particular return value indicates an error condition, the method invocation was
 2465 still successful from a perspective of the invocation operation.

2466 **Preconditions:**

- 2467 • The instance referenced by *InstancePath* shall exist in the namespace. If this is not satisfied,
2468 the operation shall fail, indicating WIPG0213.
- 2469 • The method to be invoked shall be exposed by the creation class of the instance referenced by
2470 *InstancePath*. If this is not satisfied, the operation shall fail, indicating WIPG0218.

2471 **Postconditions:**

- 2472 • The CIM method shall have been invoked.
- 2473 • Requirements on ACID properties:
- 2474 – Atomicity: Recommended
- 2475 – Update Consistency: Recommended
- 2476 – Isolation: Recommended
- 2477 – Durability: Required

2478 **Error Messages:**

2479

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0240	WBEM service limits are exceeded	Optional	Infrastructure, class implem.	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0229	Method invocation not supported by WBEM service 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.	

2480 **6.7 Class operations**

2481 This subclause defines class operations (operations that target a single CIM class or create a CIM class).
 2482 These operations include dealing with qualifier values defined on classes.

2483 **6.7.1 GetClass**

2484 **Purpose:**

2485 Retrieve a CIM class.

2486 **Operation Input Parameters:**

2487

Generic Name	Generic Type	Requirement	Description
ClassPath	ClassPath	Mandatory	Class path of the CIM class (Context Parameter)
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	NULL, or unordered set of property names to be included, acting as a restricting filter on the properties included in the returned class

2488 **Operation Output Parameters:**

2489

Generic Name	Generic Type	Requirement	Description
Class	ClassSpecification-WithPath	Mandatory	Representation of the CIM class and its class path

2490 **Description:**

2491 The *GetClass* operation retrieves a representation of the CIM class referenced by *ClassPath*.

2492 The set of properties to be included in the retrieved class shall be determined using the following
 2493 algorithm:

- 2494 • Initially, the set of properties to be included is the set of properties exposed by the class to
 2495 be retrieved. This includes all the duplicates of any duplicate non-overridden properties.
- 2496 • If the *IncludedProperties* operation input parameter is supported by the WBEM protocol
 2497 and if its value is not NULL, it acts as a restricting filter on the properties to be included in
 2498 the returned class such that any properties exposed by the class to be retrieved that are
 2499 not named in that operation parameter are removed from the set of properties to be
 2500 included. Any duplicate or invalid property names in the *IncludedProperties* operation input
 2501 parameter shall be ignored. A non-NULL empty *IncludedProperties* list removes all
 2502 properties from the set of properties to be included.

2503 **Preconditions:**

- 2504 • The CIM class referenced by *ClassPath* shall exist in the namespace. If this is not satisfied, the
2505 operation shall fail, indicating WIPG0214.

2506 **Postconditions:**

- 2507 • The CIM class shall have been returned as defined in the Description paragraph for this
2508 operation.
- 2509 • Requirements on ACID properties:
- 2510 – Atomicity: N/A
- 2511 – Update Consistency: N/A
- 2512 – Isolation: Required
- 2513 – Durability: N/A

2514 **Error Messages:**

2515

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM service 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	

2516 **6.7.2 DeleteClass**2517 **Purpose:**

2518 Delete a CIM class.

2519 **Operation Input Parameters:**

2520

Generic Name	Generic Type	Requirement	Description
ClassPath	ClassPath	Mandatory	Class path of the CIM class to be deleted (Context Parameter)

Generic Name	Generic Type	Requirement	Description
DeleteDependents	Boolean	Optional	EXPERIMENTAL: Indicates whether dependent classes and instances are to be deleted as well

2521 **Operation Output Parameters:**

2522 None.

2523 **Description:**

2524 The *DeleteClass* operation deletes the CIM class referenced by *ClassPath*.

2525

2526 EXPERIMENTAL

2527 If the WBEM protocol supports the *DeleteDependents* operation parameter, the following rules apply:

- 2528 • If *DeleteDependents* is TRUE, any classes that depend on the class referenced by
2529 *ClassPath* in the way described below shall be deleted, and any instances of the class
2530 referenced by *ClassPath* and of any classes depending on it shall be deleted according to
2531 the rules defined for the *DeleteInstance* operation. If these rules cause the rejection of an
2532 instance deletion, the *DeleteClass* operation shall fail.
- 2533 • If *DeleteDependents* is FALSE, the *DeleteClass* operation shall fail if any classes exist that
2534 depend on the class referenced by *ClassPath* in the way described below, or if the class
2535 referenced by *ClassPath* has any instances.

2536 EXPERIMENTAL

2537 If the WBEM protocol does not support the *DeleteDependents* operation parameter, the *DeleteClass*
2538 operation shall fail if any classes exist that depend on the class referenced by *ClassPath* in the way
2539 described below, or if the class referenced by *ClassPath* has any instances.

2540 For the purpose of the *DeleteClass* operation, the following classes are considered depending on the
2541 class referenced by *ClassPath*:

- 2542 • Any subclasses of any class depending on the class referenced by *ClassPath*.
- 2543 • Any association classes referencing any class depending on the class referenced by
2544 *ClassPath*.
- 2545 • Any classes defining a method with a parameter or a return value that is
 - 2546 – a reference to any class depending on the class referenced by *ClassPath*, or
 - 2547 – an embedded instance of any class depending on the class referenced by *ClassPath*,
2548 or
 - 2549 – an embedded class depending on the class referenced by *ClassPath*.
- 2550 • Any classes defining a property that is
 - 2551 – an embedded instance of any class depending on the class referenced by *ClassPath*,
2552 or
 - 2553 – an embedded class depending on the class referenced by *ClassPath*.

2554 Any classes or instances that are automatically deleted may reside in a different CIM namespace
2555 (which may reside in a different WBEM server) than the class referenced by *ClassPath*.

2556 In case of error, the consistency requirements defined in [DSP0004](#) cannot be guaranteed, but should
2557 be attempted to be satisfied in a best effort approach. In case of error, only a subset of the elements
2558 to be deleted may have been deleted, but each element shall have either been deleted completely or
2559 not at all. Also, classes shall only be deleted if all of its instances could be deleted successfully.

2560 NOTE: In a non-transactional implementation, this requires an order of deletion that starts with those elements
2561 that do not depend on the deletion of other elements.

2562 Preconditions:

- 2563 • The CIM class referenced by *ClassPath* shall exist in the namespace. If this is not satisfied, the
2564 operation shall fail, indicating WIPG0214.

2565 **Postconditions:**

- 2566 • The CIM class referenced by *ClassPath* shall have been deleted.
- 2567 • If *DeleteDependents* was TRUE:
 - 2568 – any dependent classes and instances shall have been deleted as defined in the
 - 2569 Description paragraph for this operation, and
 - 2570 – any management profile defined implicit deletions of other CIM instances shall have
 - 2571 happened, and
 - 2572 – any management profile defined effects of the deletion of all of these CIM instances
 - 2573 on any underlying resources shall have happened.
- 2574 • The consistency requirements defined in [DSP0004](#) shall be satisfied for any classes and
- 2575 instances related to the deleted classes and instances.
- 2576 • Requirements on ACID properties:
 - 2577 – Atomicity: Required, if dependent classes and instances are handled by rejection, as
 - 2578 defined in 5.8.9. Recommended, if dependent classes and instances are handled by
 - 2579 delete propagation, as defined in 5.8.9.
 - 2580 – Update Consistency: Required, if dependent classes and instances are handled by
 - 2581 rejection, as defined in 5.8.9. Recommended, if dependent classes and instances are
 - 2582 handled by delete propagation, as defined in 5.8.9.
 - 2583 – Isolation: Required, if dependent classes and instances are handled by rejection, as
 - 2584 defined in 5.8.9. Recommended, if dependent classes and instances are handled by
 - 2585 delete propagation, as defined in 5.8.9.
 - 2586 – Durability: Required

2587 **Error Messages:**

2588

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM service 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	
WIPG0224	Class has subclasses	Mandatory	Infrastructure	
WIPG0225	Class has instances	Mandatory	Infrastructure, class implem.	

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0230	Class has referencing association classes	Mandatory	Infrastructure	
WIPG0243	Timeout	Optional	Infrastructure, class implem.	
WIPG0227	Other failure	Optional	Infrastructure, class implem.	

2589 6.7.3 ModifyClass

2590 Purpose:

2591 Change the definition of a CIM class.

2592 Operation Input Parameters:

2593

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

2594 Operation Output Parameters:

2595 None.

2596 Description:

2597 The *ModifyClass* operation changes the definition of the CIM class referenced by *ClassPath*.

2598 Within the restrictions specified in the preconditions, the definition of the class referenced by
2599 *ClassPath* is replaced with the definition specified in *ModifiedClass*, as follows:

- 2600 • Any elements previously defined in the class to be changed that are not specified in
2601 *ModifiedClass* shall be removed from the class to be changed.
- 2602 • Any elements previously defined in the class to be changed that are also specified in
2603 *ModifiedClass* shall be replaced with the definition from *ModifiedClass*.
- 2604 • Any elements not previously defined in the class to be changed that are specified in
2605 *ModifiedClass* shall be added to the class to be changed, as defined in *ModifiedClass*.

2606 Any instances whose creation class is the class referenced by *ClassPath* or one of its subclasses
2607 shall be changed to reflect the changes to the class, as follows:

- 2608 • Added properties are reflected using the rules defined in the *ModifyInstance* operation
2609 when processing a list of these new properties with their values set to their class defined
2610 default values, or NULL where no class defined default value is defined.

2611 Any other changes to the class that are compatible with the preconditions do not affect existing
2612 instances, for the following reasons:

- 2613 • A compatible removal of properties from a class can only happen for overridden properties
2614 or for properties that move to a superclass, both of which is equivalent to potential changes

- 2615 of qualifier values and the default property value. Changes of qualifier values do not affect
 2616 instances. A changed default value only affects new instances, but not existing instances.
- 2617 • A compatible change of existing property definitions can only include potential changes of
 2618 qualifier values and the default property value. Changes of qualifier values do not affect
 2619 instances. A changed default value only affects new instances, but not existing instances.
 - 2620 • A compatible change of values of class qualifiers does not affect instances of the class.
 - 2621 • A compatible change to a method definition does not affect instances of the class.

2622 **Preconditions:**

- 2623 • The CIM class referenced by *ClassPath* shall exist in the namespace. If this is not satisfied, the
 2624 operation shall fail, indicating WIPG0214.
- 2625 • The name of the class defined by *ModifiedClass* shall be the name of the class referenced by
 2626 *ClassPath*. If this is not satisfied, the operation shall fail, indicating WIPG0208.
- 2627 • If the class referenced by *ClassPath* has a superclass, the class defined by *ModifiedClass* shall
 2628 specify a superclass with the same name as that superclass. If the class referenced by
 2629 *ClassPath* has no superclass, the class defined by *ModifiedClass* shall not specify a superclass.
 2630 If this is not satisfied, the operation shall fail, indicating WIPG0226.
- 2631 • The class defined by *ModifiedClass* shall only specify elements that when applied to the class to
 2632 be modified, result in a class definition that satisfies any consistency and backward compatibility
 2633 requirements defined in [DSP0004](#). For example, qualifiers with flavor *DisableOverride* shall not
 2634 be overridden, or data types of overridden properties shall not be changed. If this is not
 2635 satisfied, the operation shall fail, indicating WIPG0231.

2636 **Postconditions:**

- 2637 • The definition of the class referenced by *ClassPath* shall have been modified as defined in the
 2638 Description paragraph for this operation.
- 2639 • Any instances of the class or its subclasses shall have been changed as defined in the
 2640 Description paragraph for this operation.
- 2641 • The consistency and backward compatibility requirements defined in [DSP0004](#) shall be satisfied
 2642 for the modified class.
- 2643 • Requirements on ACID properties:
 - 2644 – Atomicity: Required
 - 2645 – Update Consistency: Required
 - 2646 – Isolation: Required
 - 2647 – Durability: Required

2648 **Error Messages:**

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM service 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	
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	

2650 6.7.4 CreateClass

2651 Purpose:

2652 Create a CIM class.

2653 Operation Input Parameters:

2654

Generic Name	Generic Type	Requirement	Description
NamespacePath	NamespacePath	Mandatory	Namespace path of the CIM namespace the class is to be created in (Context Parameter)
NewClass	ClassSpecification	Mandatory	Class specifying the definition of the class to be created

2655 Operation Output Parameters:

2656

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

2657 Description:

2658 The *CreateClass* operation creates a CIM class in the namespace referenced by *NamespacePath*,
2659 using the class definition specified in *NewClass*, and returns the class path of the new class.

2660 If properties or methods defined in *NewClass* are intended to override properties or methods defined
2661 in a superclass of *NewClass*, then they shall define an *OVERRIDE* qualifier in their definition in
2662 *NewClass*. The *CreateClass* operation shall not add such qualifiers automatically.

2663 Preconditions:

- 2664 • The CIM namespace referenced by *NamespacePath* shall exist. If this is not satisfied, the
2665 operation shall fail, indicating WIPG0204.

- 2666 • The CIM class to be created shall not exist in the namespace referenced by *NamespacePath*. If
2667 this is not satisfied, the operation shall fail, indicating WIPG0217.
- 2668 • If *NewClass* specifies a superclass, that superclass shall exist in the namespace referenced by
2669 *NamespacePath*. If this is not satisfied, the operation shall fail, indicating WIPG0226.
- 2670 NOTE: [DSP0004](#) does not provide for inheritance relationships that cross namespace boundaries.
- 2671 • The definition of *NewClass* shall satisfy any consistency requirements defined in [DSP0004](#). If
2672 this is not satisfied, the operation shall fail, indicating WIPG0208.

2673 **Postconditions:**

- 2674 • The CIM class shall have been created as defined in the Description paragraph for this
2675 operation.
- 2676 • Requirements on ACID properties:
 - 2677 – Atomicity: Required
 - 2678 – Update Consistency: Required
 - 2679 – Isolation: Required
 - 2680 – Durability: Required

2681 **Error Messages:**

2682

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM service 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	

2683 **6.8 Class enumeration operations**

2684 This subclause defines class enumeration operations (operations that enumerate CIM classes and return
2685 those classes or their class paths).

2686 **6.8.1 EnumerateClasses**2687 **Purpose:**

2688 Enumerate classes in a namespace and return these classes together with their class paths.

2689 **Operation Input Parameters:**
2690

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
IncludeInherited-Elements	boolean	Mandatory	Indicates whether any elements inherited from superclasses of <i>ClassName</i> are to be included in the returned 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

2691 **Operation Output Parameters:**
2692

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

2693 **Description:**

2694 The *EnumerateClasses* operation enumerates classes (including association and indication classes)
 2695 in the namespace specified in *NamespacePath* and returns these classes together with their class
 2696 paths.

2697 *ClassName* and *IncludeSubclasses* together determine the set of classes in the result set. The set of
 2698 classes in the result set is determined using the following algorithm:

- 2699 1) *ClassName* is optional to be specified by the WBEM client (Note that *ClassName* is
 2700 mandatory to be supported by the WBEM protocol). If *ClassName* is not specified, the
 2701 result set initially contains all top classes (that is, classes that do not have a superclass) in
 2702 the namespace. If *ClassName* is specified, the result set initially contains the subclasses of
 2703 the class specified in *ClassName* (not including the class specified in *ClassName*).

2704 2) If *IncludeSubclasses* is TRUE, then all direct and indirect subclasses of the classes that
 2705 are so far in the result set are added to the result set. Otherwise, the result set is not
 2706 changed.

2707 If *IncludeInheritedElements* is TRUE, then the set of CIM elements in each returned class shall
 2708 consist of all elements exposed by that class. Otherwise, the set of CIM elements in each returned
 2709 class shall consist only of all elements defined in the class specified in *ClassName* (including
 2710 overriding elements).

2711 The consistency model defined in 5.8 applies.

2712 **Preconditions:**

- 2713 • The CIM namespace referenced by *NamespacePath* shall exist. If this is not satisfied, the
 2714 operation shall fail, indicating WIPG0204.
- 2715 • If *ClassName* is specified, the specified CIM class shall exist in the namespace referenced by
 2716 *NamespacePath*. If this is not satisfied, the operation shall fail, indicating WIPG0214.

2717 **Postconditions:**

- 2718 • The enumerated classes with their class paths shall have been returned as defined in the
 2719 Description paragraph for this operation.
- 2720 • Requirements on ACID properties:
 - 2721 – Atomicity: N/A
 - 2722 – Update Consistency: N/A
 - 2723 – Isolation: Required at the level of single classes, as defined in 5.8.
 - 2724 – Durability: N/A

2725 **Error Messages:**

2726

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0240	WBEM service limits are exceeded	Optional	Infrastructure	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM service 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	

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0227	Other failure	Optional	Infrastructure	

2727 6.8.2 EnumerateClassNames

2728 Purpose:

2729 Enumerate classes in a namespace and return their class names.

2730 Operation Input Parameters:

2731

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

2732 Operation Output Parameters:

2733

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

2734 Description:

2735 The *EnumerateClassNames* operation enumerates classes (including association and indication
2736 classes) in the namespace specified in *NamespacePath* and returns the class names of these
2737 classes (that is, just the class name, not the class path).

2738 *ClassName* and *IncludeSubclasses* together determine the set of class names in the result set. The
2739 set of class names in the result set is determined using the following algorithm:

- 2740 1) *ClassName* is optional to be specified by the WBEM client (Note that *ClassName* is
2741 mandatory to be supported by the WBEM protocol). If *ClassName* is not specified, the
2742 result set initially contains the names of all top classes (that is, classes that do not have a
2743 superclass) in the namespace. If *ClassName* is specified, the result set initially contains the
2744 names of the subclasses of the class specified in *ClassName* (not including the class name
2745 specified in *ClassName*).
- 2746 2) If *IncludeSubclasses* is True, the class names of all direct and indirect subclasses of the
2747 classes that are so far in the result set are added to the result set. Otherwise, the result set
2748 is not changed.

2749 The consistency model defined in 5.8 applies.

2750 **Preconditions:**

- 2751 • The CIM namespace referenced by *NamespacePath* shall exist. If this is not satisfied, the
2752 operation shall fail, indicating WIPG0204.
- 2753 • If *ClassName* is specified, the specified CIM class shall exist in the namespace referenced by
2754 *NamespacePath*. If this is not satisfied, the operation shall fail, indicating WIPG0214.

2755 **Postconditions:**

- 2756 • The class names of the enumerated classes shall have been returned as defined in the
2757 Description paragraph for this operation.
- 2758 • Requirements on ACID properties:
 - 2759 – Atomicity: N/A
 - 2760 – Update Consistency: N/A
 - 2761 – Isolation: Required at the level of single classes, as defined in 5.8.
 - 2762 – Durability: N/A

2763 **Error Messages:**

2764

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0240	WBEM service limits are exceeded	Optional	Infrastructure	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM service 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	

2771 **6.8.3 AssociatorClasses**2772 **Purpose:**

2773 Enumerate the classes that are associated with a given source class and return those classes
2774 together with their class paths.

2775 **Operation Input Parameters:**
2776

Generic Name	Generic Type	Requirement	Description
ClassPath	ClassPath	Mandatory	Class path of the CIM class from which the traversal is started (the starting class) (Context Parameter)
AssociationClass-Name	ClassName	Mandatory	NULL, or name of the association class, acting as a restricting filter on the associated classes
AssociatedClass-Name	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
RoleName	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	NULL, or unordered set of property names to be included, acting as a restricting filter on the properties included in the returned class

2777 **Operation Output Parameters:**
2778

Generic Name	Generic Type	Requirement	Description
ClassList	ClassSpecification-WithPath []	Mandatory	Sequence of the associated classes with their class paths

2779 **Description:**

2780 The *AssociatorClasses* operation traverses an association from a class on a starting end to classes
2781 on all of its far ends and returns the associated CIM classes together with their class paths.

2782 The set of associated classes to be returned shall be determined using the following algorithm:

- 2783 • Initially, the set of classes to be returned is the set of all classes associated to any of the
2784 far ends of all associations referencing the starting class.

- 2785
- 2786
- 2787
- 2788
- 2789
- 2790
- If the *AssociationClassName* operation input parameter is not NULL, it acts as a restricting filter on the classes to be returned 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 returned. There shall be no validity checking performed for the *AssociationClassName* operation input parameter.
- 2791
- 2792
- 2793
- 2794
- 2795
- If the *AssociatedClassName* operation input parameter is not NULL, it acts as a restricting filter on the classes to be returned 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 returned. There shall be no validity checking performed for the *AssociatedClassName* operation input parameter.
- 2796
- 2797
- 2798
- 2799
- 2800
- If the *RoleName* operation input parameter is not NULL, it acts as a restricting filter on the classes to be returned 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 *RoleName*, is removed from the set of classes to be returned. There shall be no validity checking performed for the *RoleName* operation input parameter.
- 2801
- 2802
- 2803
- 2804
- 2805
- 2806
- If the *AssociatedRoleName* operation input parameter is not NULL, it acts as a restricting filter on the classes to be returned 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 returned. There shall be no validity checking performed for the *AssociatedRoleName* operation input parameter.

2807 The consistency model defined in 5.8 applies.

2808 The set of properties to be included in each returned associated class shall be determined using the
2809 following algorithm:

- 2810
- 2811
- 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.
- 2812
- 2813
- 2814
- 2815
- 2816
- 2817
- 2818
- 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.

2819 **Preconditions:**

- 2820
- 2821
- The CIM class referenced by *ClassPath* shall exist in the namespace. If this is not satisfied, the operation shall fail, indicating WIPG0214.
- 2822
- 2823
- 2824
- The *IncludedProperties* operation parameter, if supported by the WBEM protocol, shall only be specified with a non-NULL value if the *AssociatedClassName* operation input parameter is also non-NULL. If this is not satisfied, the operation shall fail, indicating WIPG0208.

2825 NOTE: Specifying a non-NULL value for *AssociatedClassName* ensures that the associated classes have the
2826 class specified in *AssociatedClassName* as a common superclass.

2827 **Postconditions:**

- 2828
- 2829
- The associated classes with their class paths shall have been returned as described in the Description paragraph for this operation.
- 2830
- Requirements on ACID properties:

- 2831 – Atomicity: N/A
- 2832 – Update Consistency: N/A
- 2833 – Isolation: Required at the level of single classes, as defined in 5.8.
- 2834 – Durability: N/A

2835 **Error Messages:**

2836

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0240	WBEM service limits are exceeded	Optional	Infrastructure	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM service 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	

2837 **6.8.4 AssociatorClassPaths**

2838 **Purpose:**

2839 Enumerate the classes that are associated with a given source class and return their class paths.

2840 **Operation Input Parameters:**

2841

Generic Name	Generic Type	Requirement	Description
ClassPath	ClassPath	Mandatory	Class path of the CIM class from which the traversal is started (the starting class) (Context Parameter)
AssociationClass-Name	ClassName	Mandatory	NULL, or name of the association class, acting as a restricting filter on the associated classes
AssociatedClass-Name	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

Generic Name	Generic Type	Requirement	Description
RoleName	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

2842 **Operation Output Parameters:**
2843

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

2844 **Description:**

2845 The *AssociatorClassPaths* operation traverses an association from a class on a starting end to
2846 classes on all of its far ends and returns the class paths of the associated CIM classes.

2847 The set of associated classes to be returned shall be determined using the following algorithm:

- 2848 • Initially, the set of classes to be returned is the set of all classes associated to any of the
2849 far ends of all associations referencing the starting class.
- 2850 • If the *AssociationClassName* operation input parameter is not NULL, it acts as a restricting
2851 filter on the classes to be returned such that each class that is associated with the starting
2852 class using an association class where the class or one of its superclasses does not have
2853 the name specified in *AssociationClassName*, is removed from the set of classes to be
2854 returned. There shall be no validity checking performed for the *AssociationClassName*
2855 operation input parameter.
- 2856 • If the *AssociatedClassName* operation input parameter is not NULL, it acts as a restricting
2857 filter on the classes to be returned such that each class where the class or one of its
2858 superclasses does not have the name specified in *AssociatedClassName*, is removed from
2859 the set of classes to be returned. There shall be no validity checking performed for the
2860 *AssociatedClassName* operation input parameter.
- 2861 • If the *RoleName* operation input parameter is not NULL, it acts as a restricting filter on the
2862 classes to be returned such that each class that is associated with the starting class using
2863 an association class that has a role name on its starting end that is not the role name
2864 specified in *RoleName*, is removed from the set of classes to be returned. There shall be
2865 no validity checking performed for the *RoleName* operation input parameter.
- 2866 • If the *AssociatedRoleName* operation input parameter is not NULL, it acts as a restricting
2867 filter on the classes to be returned such that each class that is associated with the starting
2868 class using an association class that has a role name on the far end referencing that class
2869 that is not the role name specified in *AssociatedRoleName*, is removed from the set of
2870 classes to be returned. There shall be no validity checking performed for the
2871 *AssociatedRoleName* operation input parameter.

2872 The consistency model defined in 5.8 applies.

2873 **Preconditions:**

- 2874 • The CIM class referenced by *ClassPath* shall exist. If this is not satisfied, the operation shall fail,
2875 indicating WIPG0214.

2876 **Postconditions:**

- 2877 • The class paths of the associated classes shall have been returned as described in the
2878 Description paragraph for this operation.
- 2879 • Requirements on ACID properties:
- 2880 – Atomicity: N/A
- 2881 – Update Consistency: N/A
- 2882 – Isolation: Required at the level of single classes, as defined in 5.8.
- 2883 – Durability: N/A

2884 **Error Messages:**

2885

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0240	WBEM service limits are exceeded	Optional	Infrastructure	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM service 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	

2886 **6.8.5 ReferenceClasses**2887 **Purpose:**

- 2888 Enumerate the association classes that reference a given source class and return these classes
2889 together with their class paths.

2890 **Operation Input Parameters:**
2891

Generic Name	Generic Type	Requirement	Description
ClassPath	ClassPath	Mandatory	Class path of the CIM class from which the traversal is started (the starting class) (Context Parameter)
AssociationClass-Name	ClassName	Mandatory	NULL, or name of the association class, acting as a restricting filter on the association classes
AssociatedClass-Name	ClassName	Mandatory	NULL, or name of the associated class on any far end of the association, acting as a restricting filter on the association classes
RoleName	PropertyName	Mandatory	NULL, or name of the role on the starting end of the association, acting as a restricting filter on the association classes
AssociatedRoleName	PropertyName	Mandatory	NULL, or name of the role on any far 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	NULL, or unordered set of property names to be included, acting as a restricting filter on the properties included in the returned classes

2892 **Operation Output Parameters:**
2893

Generic Name	Generic Type	Requirement	Description
ClassList	ClassSpecification-WithPath []	Mandatory	Sequence of the CIM association classes

2894 **Description:**

2895 The *ReferenceClasses* operation traverses an association from a class on a starting end to classes
2896 on all of its far ends and returns the CIM association classes traversed together with their class
2897 paths.

2898 The set of association classes to be returned shall be determined using the following algorithm:

- 2899 • Initially, the set of classes to be returned is the set of all association classes referencing
2900 the starting class.
- 2901 • If the *AssociationClassName* operation input parameter is not NULL, it acts as a restricting
2902 filter on the classes to be returned such that each association class where the class or one
2903 of its superclasses does not have the name specified in *AssociationClassName*, is
2904 removed from the set of classes to be returned. There shall be no validity checking
2905 performed for the *AssociationClassName* operation input parameter.

- 2906
- 2907
- 2908
- 2909
- 2910
- 2911
- If the *AssociatedClassName* operation input parameter is not NULL, it acts as a restricting filter on the classes to be returned such that each association class that has a set of references on its far ends such that none of these classes or their superclasses have the name specified in *AssociatedClassName*, is removed from the set of classes to be returned. There shall be no validity checking performed for the *AssociatedClassName* operation input parameter.
- 2912
- 2913
- 2914
- 2915
- 2916
- If the *RoleName* operation input parameter is not NULL, it acts as a restricting filter on the classes to be returned such that each association class that has a role name on its starting end that is not the role name specified in *RoleName*, is removed from the set of classes to be returned. There shall be no validity checking performed for the *RoleName* operation input parameter.
- 2917
- 2918
- 2919
- 2920
- 2921
- If the *AssociatedRoleName* operation input parameter is not NULL, it acts as a restricting filter on the classes to be returned such that each association class that has a set of role names on its far ends such that none of them is the role name specified in *AssociatedRoleName*, is removed from the set of classes to be returned. There shall be no validity checking performed for the *AssociatedRoleName* operation input parameter.

2922 The consistency model defined in 5.8 applies.

2923 The set of properties to be included in each returned association class shall be determined using the
2924 following algorithm:

- 2925
- 2926
- 2927
- Initially, the set of properties to be included is the set of properties exposed by the association class. This includes all the duplicates of any duplicate non-overridden properties.
- 2928
- 2929
- 2930
- 2931
- 2932
- 2933
- 2934
- 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.

2935 **Preconditions:**

- 2936
- 2937
- The CIM class referenced by *ClassPath* shall exist in the namespace. If this is not satisfied, the operation shall fail, indicating WIPG0214.
- 2938
- 2939
- 2940
- 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.

2941 NOTE: Specifying a non-NULL value for *AssociationClassName* ensures that the association classes have the
2942 class specified in *AssociationClassName* as a common superclass.

2943 **Postconditions:**

- 2944
- 2945
- The association classes with their class paths shall have been returned as described in the Description paragraph for this operation.
- 2946
- 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
- 2947
- 2948
- 2949
- 2950

2951
2952

Error Messages:

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0240	WBEM service limits are exceeded	Optional	Infrastructure	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM service 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	

2953 **6.8.6 ReferenceClassPaths**

2954 **Purpose:**

2955 Enumerate the association classes that reference a given source class and return their class paths.

2956 **Operation Input Parameters:**

2957

Generic Name	Generic Type	Requirement	Description
ClassPath	ClassPath	Mandatory	Class path of the CIM class from which the traversal is started (the starting class) (Context Parameter)
AssociationClass-Name	ClassName	Mandatory	NULL, or name of the association class, acting as a restricting filter on the association classes
AssociatedClass-Name	ClassName	Mandatory	NULL, or name of the associated class on any far end of the association, acting as a restricting filter on the association classes
RoleName	PropertyName	Mandatory	NULL, or name of the role on the starting end of the association, acting as a restricting filter on the association classes
AssociatedRoleName	PropertyName	Mandatory	NULL, or name of the role on any far end of the association, acting as a restricting filter on the association classes

2958 **Operation Output Parameters:**
2959

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

2960 **Description:**

2961 The *ReferenceClassPaths* operation traverses an association from a class on a starting end to
2962 classes on all of its far ends and returns the class paths of the CIM association classes traversed.

2963 The set of association classes to be returned shall be determined using the following algorithm:

- 2964 • Initially, the set of classes to be returned is the set of all association classes referencing
2965 the starting class.
- 2966 • If the *AssociationClassName* operation input parameter is not NULL, it acts as a restricting
2967 filter on the classes to be returned such that each association class where the class or one
2968 of its superclasses does not have the name specified in *AssociationClassName*, is
2969 removed from the set of classes to be returned. There shall be no validity checking
2970 performed for the *AssociationClassName* operation input parameter.
- 2971 • If the *AssociatedClassName* operation input parameter is not NULL, it acts as a restricting
2972 filter on the classes to be returned such that each association class that has a set of
2973 references on its far ends such that none of these classes or their superclasses have the
2974 name specified in *AssociatedClassName*, is removed from the set of classes to be
2975 returned. There shall be no validity checking performed for the *AssociatedClassName*
2976 operation input parameter.
- 2977 • If the *RoleName* operation input parameter is not NULL, it acts as a restricting filter on the
2978 classes to be returned such that each association class that has a role name on its starting
2979 end that is not the role name specified in *RoleName*, is removed from the set of classes to
2980 be returned. There shall be no validity checking performed for the *RoleName* operation
2981 input parameter.
- 2982 • If the *AssociatedRoleName* operation input parameter is not NULL, it acts as a restricting
2983 filter on the classes to be returned such that each association class that has a set of role
2984 names on its far ends such that none of them is the role name specified in
2985 *AssociatedRoleName*, is removed from the set of classes to be returned. There shall be no
2986 validity checking performed for the *AssociatedRoleName* operation input parameter.

2987 The consistency model defined in 5.8 applies.

2988 **Preconditions:**

- 2989 • The CIM class referenced by *ClassPath* shall exist. If this is not satisfied, the operation shall fail,
2990 indicating WIPG0214.

2991 **Postconditions:**

- 2992 • The association classes with their class paths shall have been returned as described in the
2993 Description paragraph for this operation.
- 2994 • Requirements on ACID properties:
 - 2995 – Atomicity: N/A
 - 2996 – Update Consistency: N/A
 - 2997 – Isolation: Required at the level of single classes, as defined in 5.8.

2998 – Durability: N/A

2999 **Error Messages:**

3000

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0240	WBEM service limits are exceeded	Optional	Infrastructure	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM service 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	

3001 **6.9 Qualifier type operations**

3002 This subclause defines operations that deal with qualifier types. As defined in [DSP0004](#), qualifier types
 3003 represent the declarations of qualifiers, not their values.

3004 **6.9.1 GetQualifierType**

3005 **Purpose:**

3006 Retrieve a qualifier type.

3007 **Operation Input Parameters:**

3008

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

3009 **Operation Output Parameters:**

3010

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

3011 **Description:**3012 The *GetQualifierType* operation retrieves the CIM qualifier type referenced by *QualifierTypePath*.3013 **Preconditions:**

- 3014 • The CIM qualifier type referenced by
- QualifierTypePath*
- shall exist in the namespace. If this is
-
- 3015 not satisfied, the operation shall fail, indicating WIPG0215.

3016 **Postconditions:**

- 3017 • The qualifier type shall have been returned as described in the Description paragraph for this
-
- 3018 operation.
-
- 3019 • Requirements on ACID properties:
-
- 3020 – Atomicity: N/A
-
- 3021 – Update Consistency: N/A
-
- 3022 – Isolation: Required
-
- 3023 – Durability: N/A

3024 **Error Messages:**

3025

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM service 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	

3026 **6.9.2 DeleteQualifierType**3027 **Purpose:**

3028 Delete a qualifier type.

3029 **Operation Input Parameters:**
3030

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

3031 **Operation Output Parameters:**

3032 None.

3033 **Description:**

3034 The *DeleteQualifierType* operation deletes the CIM qualifier type referenced by *QualifierTypePath*.

3035 As defined in [DSP0004](#), any namespace needs to contain qualifier types for the meta qualifiers and
3036 standard qualifiers, and may contain qualifier types for the optional qualifiers. Thus, deleting any
3037 required qualifier types from a namespace will render that namespace non-compliant to [DSP0004](#).

3038 **Preconditions:**

- 3039 • The CIM qualifier type referenced by *QualifierTypePath* shall exist in the namespace. If this is
3040 not satisfied, the operation shall fail, indicating WIPG0215.
- 3041 • The qualifier identified by *QualifierTypePath* shall not be specified on any element in the same
3042 namespace. If this is not satisfied, the operation shall fail, indicating WIPG0233.

3043 **Postconditions:**

- 3044 • The CIM qualifier type shall have been deleted as described in the Description paragraph for
3045 this operation.
- 3046 • Requirements on ACID properties:
 - 3047 – Atomicity: Required
 - 3048 – Update Consistency: Required
 - 3049 – Isolation: Required
 - 3050 – Durability: Required

3051 **Error Messages:**
3052

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM service 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	
WIPG0215	Qualifier type not found	Mandatory	Infrastructure	
WIPG0233	Qualifier type is used	Mandatory	Infrastructure	
WIPG0243	Timeout	Optional	Infrastructure	
WIPG0227	Other failure	Optional	Infrastructure	

3053 6.9.3 ModifyQualifierType

3054 Purpose:

3055 Change the definition of a CIM qualifier type.

3056 Operation Input Parameters:

3057

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

3058 Operation Output Parameters:

3059 None.

3060 Description:

3061 The *ModifyQualifierType* operation changes the definition of the CIM qualifier type referenced by
3062 *QualifierTypePath*.

3063 The definition of the qualifier type referenced by *QualifierTypePath* is replaced with the definition
3064 specified in *ModifiedQualifierType*.

3065 As defined in [DSP0004](#), any namespace needs to contain qualifier types for the meta qualifiers and
3066 standard qualifiers, and may contain qualifier types for the optional qualifiers. Thus, changing these
3067 qualifier types in a namespace inconsistently with their [DSP0004](#) definition will render that
3068 namespace non-compliant to [DSP0004](#).

3069 Preconditions:

- 3070 • The CIM qualifier type referenced by *QualifierTypePath* shall exist in the namespace. If this is
3071 not satisfied, the operation shall fail, indicating WIPG0215.
- 3072 • The name of the qualifier type defined by *ModifiedQualifierType* shall be the name of the
3073 qualifier type referenced by *QualifierTypePath*. If this is not satisfied, the operation shall fail,
3074 indicating WIPG0208.
- 3075 • The request to modify the qualifier type shall satisfy any backward compatibility requirements
3076 defined in [DSP0004](#). If this is not satisfied, the operation shall fail, indicating WIPG0234.

- 3077 • If the qualifier type referenced by *QualifierTypePath* is one of the qualifiers defined in [DSP0004](#),
3078 (i.e., meta, standard, and optional qualifiers), the new definition of the qualifier in
3079 *ModifiedQualifierType* shall be consistent with the definition of the qualifier in [DSP0004](#). If this is
3080 not satisfied, the operation shall fail, indicating WIPG0245.

3081 **Postconditions:**

- 3082 • The definition of the qualifier type referenced by *QualifierTypePath* shall have been modified as
3083 defined in the Description paragraph for this operation.
- 3084 • The backward compatibility requirements defined in [DSP0004](#) shall be satisfied for the modified
3085 qualifier type.
- 3086 • Requirements on ACID properties:
 - 3087 – Atomicity: Required
 - 3088 – Update Consistency: Required
 - 3089 – Isolation: Required
 - 3090 – Durability: Required

3091 **Error Messages:**

3092

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM service 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	

3093 **6.9.4 CreateQualifierType**

3094 **Purpose:**

3095 Create a CIM qualifier type.

3096 **Operation Input Parameters:**
3097

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

3098 **Operation Output Parameters:**
3099

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

3100 **Description:**

3101 The *CreateQualifierType* operation creates a CIM qualifier type in the namespace referenced by
3102 *NamespacePath*, using the qualifier type definition specified in *NewQualifierType*, and returns the
3103 qualifier type path of the new qualifier type.

3104 As defined in [DSP0004](#), any namespace needs to contain qualifier types for the meta qualifiers and
3105 standard qualifiers, and may contain qualifier types for the optional qualifiers. Thus, creating these
3106 qualifier types in a namespace inconsistently with their [DSP0004](#) definition will render that
3107 namespace non-compliant to [DSP0004](#).

3108 **Preconditions:**

- 3109 • The CIM namespace referenced by *NamespacePath* shall exist. If this is not satisfied, the
3110 operation shall fail, indicating WIPG0204.
- 3111 • The CIM qualifier type to be created shall not exist in the namespace referenced by
3112 *NamespacePath*. If this is not satisfied, the operation shall fail, indicating WIPG0208.
- 3113 • If the qualifier type defined in *NewQualifierType* is one of the qualifiers defined in [DSP0004](#),
3114 (i.e., meta, standard, and optional qualifiers), the definition of the qualifier in *NewQualifierType*
3115 shall be consistent with the definition of the qualifier in [DSP0004](#). If this is not satisfied, the
3116 operation shall fail, indicating WIPG0245.

3117 **Postconditions:**

- 3118 • The CIM qualifier type shall have been created as defined in the Description paragraph for this
3119 operation.
- 3120 • Requirements on ACID properties:
 - 3121 – Atomicity: Required
 - 3122 – Update Consistency: Required
 - 3123 – Isolation: Required
 - 3124 – Durability: Required

3125 **Error Messages:**
3126

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM service 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	

3127 **6.9.5 EnumerateQualifierTypes**

3128 **Purpose:**

3129 Enumerate the qualifier types in a namespace.

3130 **Operation Input Parameters:**
3131

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

3132 **Operation Output Parameters:**
3133

Generic Name	Generic Type	Requirement	Description
QualifierTypeList	QualifierTypeWith-Path []	Mandatory	Sequence of the enumerated CIM qualifier types with their qualifier type paths

3134 **Description:**

3135 The *EnumerateQualifierTypes* operation enumerates all CIM qualifier types in the namespace
3136 referenced by *NamespacePath*, and returns these qualifier types together with their qualifier type
3137 paths.

3138 **Preconditions:**

- 3139 • The CIM namespace referenced by *NamespacePath* shall exist. If this is not satisfied, the
3140 operation shall fail, indicating WIPG0204.

3141 **Postconditions:**

- 3142 • The CIM qualifier types with their qualifier type paths shall have been enumerated as defined in
3143 the Description paragraph for this operation.
- 3144 • Requirements on ACID properties:
- 3145 – Atomicity: N/A
- 3146 – Update Consistency: N/A
- 3147 – Isolation: Required at the level of single qualifier types, as defined in 5.8.
- 3148 – Durability: N/A

3149 **Error Messages:**

3150

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0240	WBEM service limits are exceeded	Optional	Infrastructure	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM service 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	

ANNEX A (informative)

Future operations

3155 This annex provides ideas for future operations or extensions to existing operations.

3156 **A.1 Test for property modifiability**

3157 Today, management profiles specify the modifiability of properties or an algorithm how to find out their
3158 modifiability at runtime. Usually, this includes the overhead of capability based mechanisms most of the
3159 time at the level of single properties. Because of this overhead, it is defined rarely in profiles and thus left
3160 to be decided by the implementation, with no defined way for a client to find out about it upfront.

3161 An operation (or an extension to an existing operation) that allows testing for modifiability of properties in
3162 a consistent way without depending on hard wired understanding of profile defined modifiability or profile
3163 defined algorithms to find out modifiability would be a worthwhile extension.

3164 **A.2 Retrieval of associated instance graph**

3165 Today, a graph of associated instances can be retrieved only piece by piece, even distinguishing between
3166 retrieval of association instances (e.g., via *GetReferencingInstance...*) and associated instances (e.g., via
3167 *GetAssociatedInstance...*). Also, retrieving the associated instances associated by different associations
3168 may involve the invocation of multiple class implementations in typical CIMOM/provider based
3169 implementations, which could be optimized by having a single implementation of a more complex
3170 operation like the one proposed here.

3171 An operation would be helpful that can retrieve the graph of associated instances including their
3172 associations. Ideally, the operation would be able to traverse multiple association hops in one invocation.

3173 One possible definition of such operations could be:

3174 Direct retrieval: The *GetAssociatedGraphInstancesWithPath* operation traverses an association from an
3175 instance on a source end to instances on all of its far ends and returns the associated instances and their
3176 association instances, each together with their instance paths. This operation can be used to return one
3177 set of instances that would have otherwise required at least twice as many operations (one set to get the
3178 associations and another to get the related instances).

3179 Pulled retrieval: The *OpenAssociatedGraphInstancesWithPath* operation establishes and opens an
3180 enumeration session for enumerating instances that are associated with the specified source instance,
3181 and their association instances, including their instance paths. This operation can be used to return one
3182 set of instances that would have otherwise required at least twice as many operations (one set to get the
3183 associations and another to get the related instances).

ANNEX B (informative)

3184
3185
3186
3187

Changed generic operation names

3188 Version 1.0.2 of this document changed the names of the generic operations in order to align them with
3189 the operation names of the CIM-XML protocol (see [DSP0200](#)) that are used in current management
3190 profiles. This change allows management profiles to more easily use the generic operation names, which
3191 is required when using the condensed format defined in [DSP1001](#) V1.1 or when migrating profiles to
3192 become machine readable (see [DSP2023](#)).

3193 Note that the new generic operations are not always 1:1 with the CIM-XML operations: For example, in
3194 CIM-XML, the association operations are one set of operations covering both instance and class level,
3195 while in generic operations, class and instance level operations continue to be separated.

3196 This name change is incompatible for management profiles that specified operation requirements using
3197 the old generic operation names. There is only one such DMTF profile ([DSP1054 Version 1.2](#)). However,
3198 it is not an incompatible change for implementations of such profiles because the names of generic
3199 operations are not visible in the implementation; they remain at the specification level. All APIs and
3200 protocols the DMTF knows about do not currently use the generic operation names in their specifications
3201 or in their implementations (except for mappings between the APIs or protocols and generic operations).

3202 Table 2 lists the old and new operation names for all operations defined in this document.

3203

Table 2 – Changed generic operation names

New operation name (starting with V1.0.2)	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	GetClassInstancesWithPath	yes	See 6.4.1
EnumerateInstanceNames	GetClassInstancePaths	yes	See 6.4.2
Associators	GetAssociatedInstancesWithPath	yes	See 6.4.3
AssociatorNames	GetAssociatedInstancePaths	yes	See 6.4.4
References	GetReferencingInstancesWithPath	yes	See 6.4.5
ReferenceNames	GetReferencingInstancePaths	yes	See 6.4.6
OpenEnumerateInstances	OpenClassInstancesWithPath	yes	See 6.5.3
OpenEnumerateInstancePaths	OpenClassInstancePaths	yes	See 6.5.4
OpenAssociators	OpenAssociatedInstancesWithPath	yes	See 6.5.5
OpenAssociatorPaths	OpenAssociatedInstancePaths	yes	See 6.5.6
OpenReferences	OpenReferencingInstancesWithPath	yes	See 6.5.7
OpenReferencePaths	OpenReferencingInstancePaths	yes	See 6.5.8
OpenQueryInstances	OpenQueryInstances	no	See 6.5.9

New operation name (starting with V1.0.2)	Old operation name (in V1.0.0 and V1.0.1)	Name Changed	Description
PullInstancesWithPath	PullInstancesWithPath	no	See 6.5.11
PullInstancePaths	PullInstancePaths	no	See 6.5.12
PullInstances	PullInstances	no	See 6.5.13
CloseEnumeration	CloseEnumeration	no	See 6.5.14
EnumerationCount	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

3204 Notes:

- 3205 (1) The old *GetTopClassesWithPath* and *GetSubClassesWithPath* operations have been merged
3206 into the new *EnumerateClasses* operation that covers both top classes and subclasses.
3207 (2) The old *GetTopClassPaths* and *GetSubClassPaths* operations have been merged into the new
3208 *EnumerateClassNames* operation that covers both top classes and subclasses.

ANNEX C
(informative)**Change log**3209
3210
3211
3212

3213

Version	Date	Description
1.0.0	2010-04-22	
1.0.1	2012-08-30	Published as DMTF Standard, with the following changes: <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.
1.0.2	2013-10-22	Member review of DMTF Draft Standard, with the following changes: <ul style="list-style-type: none">• Errata: Changed the names of the generic operations to be aligned with the CIM-XML operation names. See ANNEX B for details.• 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.

Bibliography

3214

3215 DMTF DSP0200, *CIM Operations over HTTP 1.3*,
3216 http://www.dmtf.org/standards/published_documents/DSP0200_1.3.pdf

3217 DMTF DSP0201, *Representation of CIM in XML 2.3*,
3218 http://www.dmtf.org/standards/published_documents/DSP0201_2.3.pdf

3219 DMTF DSP0202, *CIM Query Language Specification 1.0*,
3220 http://www.dmtf.org/standards/published_documents/DSP0202_1.0.pdf

3221 DMTF DSP0203, *DTD for Representation of CIM in XML 2.3*,
3222 http://www.dmtf.org/standards/published_documents/DSP0203_2.3.dtd

3223 DMTF DSP0214, *Server Management Command Line Protocol Specification 1.0*,
3224 http://www.dmtf.org/standards/published_documents/DSP0214_1.0.pdf

3225 DMTF DSP0226, *Web Services for Management 1.0*,
3226 http://www.dmtf.org/standards/published_documents/DSP0226_1.0.pdf

3227 DMTF DSP0227, *WS-Management CIM Binding Specification 1.0*,
3228 http://www.dmtf.org/standards/published_documents/DSP0227_1.0.pdf

3229 DMTF DSP0230, *WS-CIM Mapping Specification 1.0*,
3230 http://www.dmtf.org/standards/published_documents/DSP0230_1.0.pdf

3231 DMTF DSP1001, *Management Profile Specification Usage Guide 1.1*,
3232 http://www.dmtf.org/standards/published_documents/DSP1001_1.1.pdf

3233 DMTF DSP1054, *Indications Profile 1.2*,
3234 http://www.dmtf.org/standards/published_documents/DSP1054_1.2.pdf

3235 DMTF DSP2023, *XML Management Profile Setup and Samples 1.0*,
3236 http://www.dmtf.org/standards/published_documents/DSP2023_1.0.zip

3237 JCP JSR-48, *Java Community Process JSR-48: WBEM Services Specification*,
3238 <http://jcp.org/en/jsr/detail?id=48>

3239 The Open Group CMPI, *Systems Management: Common Manageability Programming Interface 1.0*,
3240 <http://www.opengroup.org/bookstore/catalog/c051.htm>