



Document number: DSP0223

Date: 2012-08-30

Version: 1.0.1

1

2

3

4

5

6

7

8

## Generic Operations

9

10

11

12

13

14 **Document type: Specification**

15 **Document status: DMTF Standard**

16 **Document language: en-US**

17

18 Copyright notice

19 Copyright © 2007–2012 Distributed Management Task Force, Inc. (DMTF). All rights reserved.

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

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34 withdrawn or modified after publication, and shall be indemnified and held harmless by any party  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

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

## Contents

41	Foreword .....	7
42	Acknowledgements .....	7
43	Document conventions .....	7
44	Typographical conventions .....	7
45	Experimental material .....	7
46	1 Scope .....	8
47	2 Normative references .....	8
48	3 Terms and definitions .....	9
49	4 Symbols and abbreviated terms .....	12
50	5 Concepts .....	12
51	5.1 Generic operations model .....	12
52	5.2 Generic operations mappings .....	13
53	5.2.1 Overview .....	13
54	5.2.2 Recommendations .....	14
55	5.3 Conformance to generic operations .....	15
56	5.3.1 Conformance of WBEM protocols or APIs .....	15
57	5.3.2 Conformance of WBEM operations or API calls .....	15
58	5.3.3 Requirement levels for operation parameters .....	15
59	5.4 Generic types .....	16
60	5.4.1 CIM data types .....	16
61	5.4.2 NamespacePath .....	16
62	5.4.3 InstancePath .....	16
63	5.4.4 ClassPath .....	16
64	5.4.5 QualifierTypePath .....	16
65	5.4.6 InstanceSpecification .....	17
66	5.4.7 ClassSpecification .....	17
67	5.4.8 QualifierType .....	18
68	5.4.9 InstanceSpecificationWithPath .....	18
69	5.4.10 ClassSpecificationWithPath .....	18
70	5.4.11 QualifierTypeWithPath .....	19
71	5.4.12 ClassName .....	19
72	5.4.13 PropertyName .....	19
73	5.4.14 MethodName .....	19
74	5.4.15 ParameterValue .....	19
75	5.4.16 ReturnValue .....	19
76	5.4.17 QueryString .....	19
77	5.4.18 QueryLanguage .....	19
78	5.4.19 EnumerationContext .....	20
79	5.5 Success and failure .....	20
80	5.6 Preconditions and postconditions .....	20
81	5.7 Generic error messages .....	20
82	5.8 Consistency model .....	21
83	5.8.1 Definition of ACID properties .....	21
84	5.8.2 Time consistency within a CIM instance .....	22
85	5.8.3 Staleness of information returned .....	22
86	5.8.4 Isolation between operations .....	22
87	5.8.5 Duplicate return of CIM objects or object paths .....	23
88	5.8.6 Time consistency between returned CIM objects .....	23
89	5.8.7 Order of returned CIM objects .....	23
90	5.8.8 Validity of returned object paths .....	23
91	5.8.9 Effects of deleting an instance .....	24
92	6 Generic operations .....	26

93	6.1	Description format.....	27
94	6.2	Common operation parameters for all operations .....	29
95	6.2.1	IncludeClassOrigin.....	29
96	6.2.2	IncludeQualifiers .....	29
97	6.2.3	<element>List .....	29
98	6.3	Instance operations.....	30
99	6.3.1	GetInstance.....	30
100	6.3.2	DeleteInstance.....	32
101	6.3.3	ModifyInstance.....	34
102	6.3.4	CreateInstance.....	36
103	6.4	Direct instance enumeration operations .....	39
104	6.4.1	GetClassInstancesWithPath .....	39
105	6.4.2	GetClassInstancePaths .....	41
106	6.4.3	GetAssociatedInstancesWithPath .....	43
107	6.4.4	GetAssociatedInstancePaths.....	46
108	6.4.5	GetReferencingInstancesWithPath.....	49
109	6.4.6	GetReferencingInstancePaths.....	52
110	6.5	Pulled instance enumeration operations.....	55
111	6.5.1	General behavioral rules.....	55
112	6.5.2	Common operation parameters for the open operations.....	57
113	6.5.3	OpenClassInstancesWithPath .....	59
114	6.5.4	OpenClassInstancePaths .....	63
115	6.5.5	OpenAssociatedInstancesWithPath .....	66
116	6.5.6	OpenAssociatedInstancePaths.....	70
117	6.5.7	OpenReferencingInstancesWithPath.....	74
118	6.5.8	OpenReferencingInstancePaths.....	79
119	6.5.9	OpenQueryInstances .....	83
120	6.5.10	Common operation parameters for the pull operations .....	85
121	6.5.11	PullInstancesWithPath .....	86
122	6.5.12	PullInstancePaths .....	88
123	6.5.13	PullInstances.....	90
124	6.5.14	CloseEnumeration .....	92
125	6.5.15	EnumerationCount .....	94
126	6.6	Method invocation.....	95
127	6.6.1	InvokeMethod .....	96
128	6.6.2	InvokeStaticMethod .....	97
129	6.7	Class operations .....	99
130	6.7.1	GetClass .....	99
131	6.7.2	DeleteClass.....	101
132	6.7.3	ModifyClass .....	104
133	6.7.4	CreateClass .....	106
134	6.8	Class enumeration operations .....	108
135	6.8.1	GetTopClassesWithPath .....	108
136	6.8.2	GetTopClassPaths.....	110
137	6.8.3	GetSubClassesWithPath .....	111
138	6.8.4	GetSubClassPaths.....	113
139	6.8.5	GetAssociatedClassesWithPath .....	115
140	6.8.6	GetAssociatedClassPaths .....	118
141	6.8.7	GetReferencingClassesWithPath .....	120
142	6.8.8	GetReferencingClassPaths .....	122
143	6.9	Qualifier type operations.....	124
144	6.9.1	GetQualifierType.....	125
145	6.9.2	DeleteQualifierType .....	126
146	6.9.3	ModifyQualifierType .....	127
147	6.9.4	CreateQualifierType.....	129
148	6.9.5	EnumerateQualifierTypesWithPath .....	131

149	ANNEX A (informative) Future operations .....	133
150	A.1 Test for property modifiability .....	133
151	A.2 Retrieval of associated instance graph .....	133
152	ANNEX B (informative) Change log .....	134
153	Bibliography .....	136
154		
155	<b>Figures</b>	
156	Figure 1 – Generic operations model .....	13
157	Figure 2 – Generic operations mappings .....	14
158		
159	<b>Tables</b>	
160	Table 1 – List of generic operations .....	26
161		



163

## Foreword

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

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

### 168 Acknowledgements

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

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

### 176 Document conventions

#### 177 Typographical conventions

178 The following typographical conventions are used in this document:

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

#### 184 Experimental material

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

190 The following typographical convention indicates experimental material:

---

---

#### 191 **EXPERIMENTAL**

192 Experimental material appears here.

---

---

#### 193 **EXPERIMENTAL**

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

196

197

# Generic Operations

## 198 1 Scope

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

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

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

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

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

## 229 2 Normative references

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

234 DMTF DSP0004, *CIM Infrastructure Specification 2.6*,  
235 [http://www.dmtf.org/standards/published\\_documents/DSP0004\\_2.6.pdf](http://www.dmtf.org/standards/published_documents/DSP0004_2.6.pdf)

236 DMTF DSP0207, *WBEM URI Mapping 1.0*,  
237 [http://www.dmtf.org/standards/published\\_documents/DSP0207\\_1.0.pdf](http://www.dmtf.org/standards/published_documents/DSP0207_1.0.pdf)

238 DMTF DSP0228, *Message Registry XML Schema 1.1*,  
239 [http://schemas.dmtf.org/wbem/messageregistry/1/dsp0228\\_1.1.xsd](http://schemas.dmtf.org/wbem/messageregistry/1/dsp0228_1.1.xsd)

240 DMTF DSP8016, *WBEM Operations Message Registry 1.0*,  
241 [http://schemas.dmtf.org/wbem/messageregistry/1/dsp8016\\_1.0.xml](http://schemas.dmtf.org/wbem/messageregistry/1/dsp8016_1.0.xml)

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

## 244 **3 Terms and definitions**

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

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

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

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

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

### 260 **3.1**

#### 261 **class path**

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

### 264 **3.2**

#### 265 **creation class**

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

### 268 **3.3**

#### 269 **duplicate object**

270 objects in a result set that have duplicate object paths

### 271 **3.4**

#### 272 **duplicate object path**

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

- 274 **3.5**  
275 **effective qualifier value**  
276 The effective value of a qualifier specified on a schema element is the value that determines the qualifier  
277 behavior for the schema element, taking the qualifier propagation rules into account. For a complete  
278 definition, see [DSP0004](#).
- 279 **3.6**  
280 **exposed elements of a class**  
281 The set of schema elements exposed by a class (i.e., properties and methods) is the union of the set of  
282 elements defined in the class and the set of inherited elements that are not overridden in the class. For a  
283 complete definition, see [DSP0004](#).
- 284 **3.7**  
285 **generic operation**  
286 a generic operation as defined in this specification
- 287 **3.8**  
288 **generic operations mapping**  
289 a mapping of generic operations to the operations of some other protocol (e.g., WBEM operations) or to  
290 the calls of some API, as defined in 5.2
- 291 **3.9**  
292 **instance path**  
293 a special kind of object path addressing a CIM instance that is accessible through a WBEM server  
294 For details, see [DSP0004](#).
- 295 **3.10**  
296 **isolation**  
297 the set of behaviors that describe how the execution of an operation affects the execution of another,  
298 concurrent operation, as defined in 5.8.4
- 299 **3.11**  
300 **management profile**  
301 a management profile as defined in [DSP1001](#)  
302 As used in this specification, the term includes all possible owners of such profiles, including other  
303 standards organizations and vendors.
- 304 **3.12**  
305 **namespace path**  
306 a special kind of object path addressing a CIM namespace that is accessible through a WBEM server  
307 For details, see [DSP0004](#).
- 308 **3.13**  
309 **object**  
310 a class, instance, qualifier type or namespace that is accessible through a WBEM server  
311 For details, see [DSP0004](#).
- 312 **3.14**  
313 **object path**  
314 the address of an object that is accessible through a WBEM server  
315 For details, see [DSP0004](#).

- 316 **3.15**  
317 **qualifier type path**  
318 a special kind of object path addressing a CIM qualifier type that is accessible through a WBEM server  
319 For details, see [DSP0004](#).
- 320 **3.16**  
321 **volatile property**  
322 a property in a CIM instance whose value may change as a WBEM client obtains the instance repeatedly  
323 without performing any client originated updates to the property value
- 324 **3.17**  
325 **WBEM client**  
326 a CIM client (see [DSP0004](#)) that supports a WBEM protocol  
327 A WBEM client originates WBEM operations for processing by a WBEM server. This definition does not  
328 imply any particular implementation architecture or scope, such as a client library component or an entire  
329 management application. For details, see 5.1.
- 330 **3.18**  
331 **WBEM indication**  
332 an interaction within a WBEM protocol that is originated on a WBEM server and processed by a WBEM  
333 listener  
334 This release of this specification does not cover WBEM indications.
- 335 **3.19**  
336 **WBEM listener**  
337 a CIM listener (see [DSP0004](#)) that supports a WBEM protocol  
338 A WBEM listener processes WBEM indications originated by a WBEM server. This definition does not  
339 imply any particular implementation architecture or scope, such as a standalone demon component or an  
340 entire management application.  
341 This release of this specification does not cover WBEM listeners.
- 342 **3.20**  
343 **WBEM operation**  
344 an interaction within a WBEM protocol that is originated by a WBEM client and processed by a WBEM  
345 server  
346 For details, see 5.1.
- 347 **3.21**  
348 **WBEM protocol**  
349 a communications protocol between WBEM client, WBEM server and WBEM listener  
350 A WBEM protocol defines how the WBEM operations and WBEM indications work, on top of an  
351 underlying protocol layer (for example, HTTP, SOAP, or TCP). For details, see 5.1.
- 352 **3.22**  
353 **WBEM protocol mapping**  
354 a mapping of generic operations to a WBEM protocol, as defined in 5.2
- 355 **3.23**  
356 **WBEM server**  
357 a CIM server (see [DSP0004](#)) that supports a WBEM protocol  
358 A WBEM server processes WBEM operations originated by a WBEM client, and originates WBEM  
359 indications for processing by a WBEM listener. This definition does not imply any particular  
360 implementation architecture, such as a separation into a CIMOM and provider components. For details,  
361 see 5.1.

## 362 **4 Symbols and abbreviated terms**

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

### 365 **4.1**

#### 366 **API**

367 Application Programming Interface

### 368 **4.2**

#### 369 **CIM**

370 Common Information Model, defined by DMTF

### 371 **4.3**

#### 372 **CQL**

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

### 374 **4.4**

#### 375 **HTTP**

376 Hyper Text Transfer Protocol, defined by W3C

### 377 **4.5**

#### 378 **UML**

379 Unified Modeling Language, defined by OMG

### 380 **4.6**

#### 381 **WBEM**

382 Web Based Enterprise Management, defined by DMTF

### 383 **4.7**

#### 384 **XML**

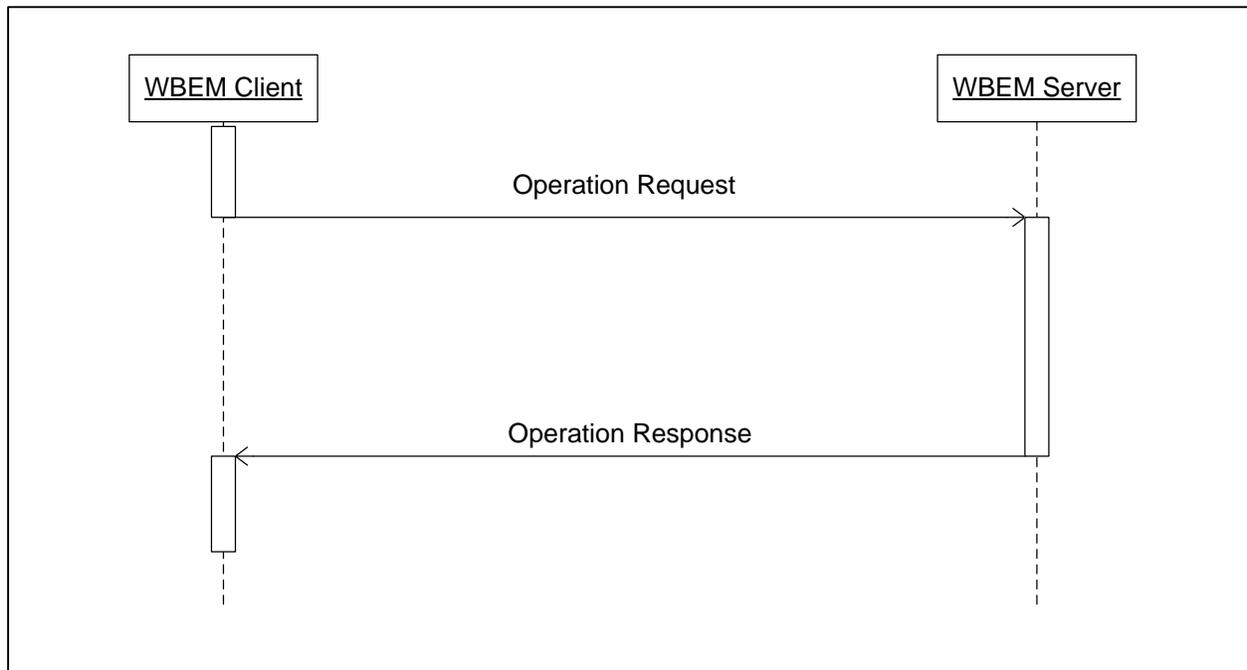
385 Extensible Markup Language, defined by W3C

## 386 **5 Concepts**

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

### 388 **5.1 Generic operations model**

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



390  
391

392

**Figure 1 – Generic operations model**

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

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

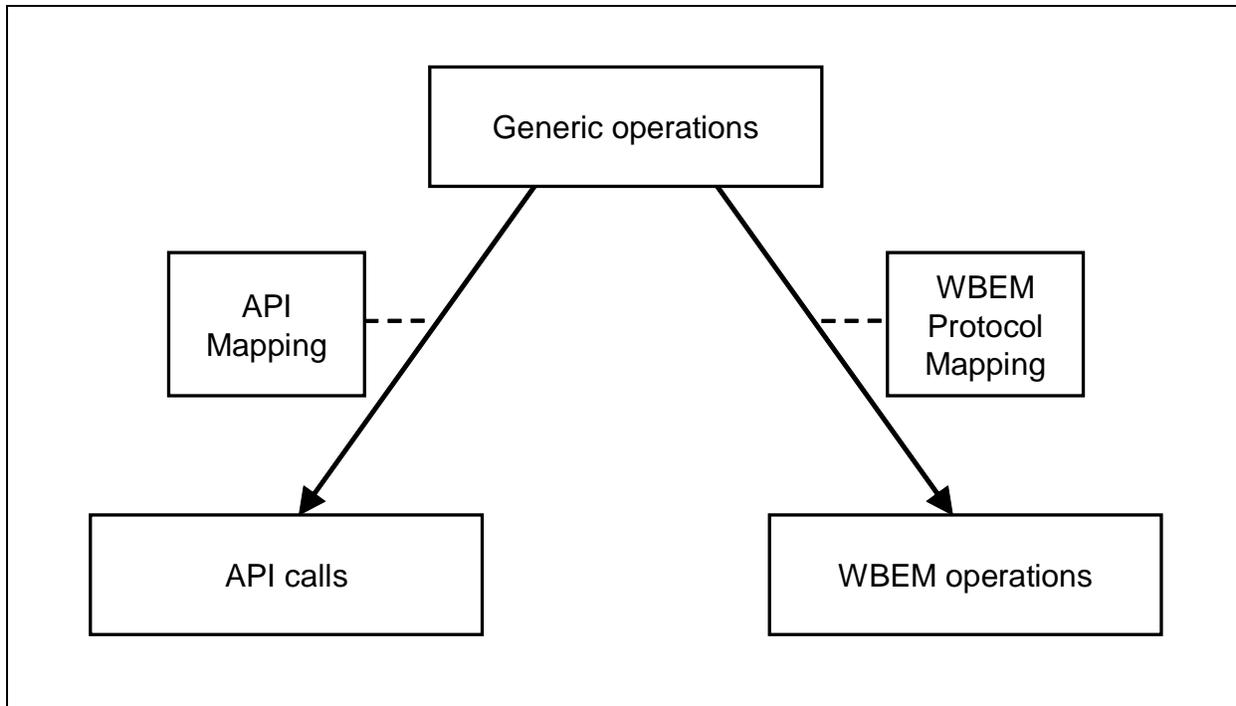
400 The operation request and operation response at the level of generic operations do not necessarily need  
401 to correspond directly to messages that are flowing at the level of the WBEM protocol. For example, the  
402 operation response may be delivered asynchronously at the level of the WBEM protocol.

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

## 407 5.2 Generic operations mappings

### 408 5.2.1 Overview

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



413

414

**Figure 2 – Generic operations mappings**

415 **5.2.2 Recommendations**

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

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

421 The following recommendations apply:

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

### 433 **5.3 Conformance to generic operations**

434 Conformance to generic operations is defined at two levels:

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

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

#### 440 **5.3.1 Conformance of WBEM protocols or APIs**

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

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

#### 445 **5.3.2 Conformance of WBEM operations or API calls**

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

- 448 • The generic operation has one or more corresponding WBEM operations or API calls that  
449 provide the functionality of the generic operation. The names of these corresponding WBEM  
450 operations or API calls may be different from the name of the generic operation.
- 451 • Functionalities that are required to be supported for a generic operation are supported by the  
452 corresponding WBEM operations or API calls with the semantics defined by the generic  
453 operation.
- 454 • If functionalities that are optional to be supported for a generic operation are supported by the  
455 corresponding WBEM operations or API calls, they are supported with the semantics defined by  
456 the generic operation.
- 457 • Each parameter of a generic operation is mapped to one or more corresponding parameters of  
458 the corresponding WBEM operations or API calls
- 459 • For each parameter of a generic operation, the provisions defined in 5.3.3 are satisfied.

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

#### 463 **5.3.3 Requirement levels for operation parameters**

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

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

##### 468 **Mandatory**

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

473       **Conditional**

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

478       **Optional**

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

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

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

489       **5.4 Generic types**

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

492       **5.4.1 CIM data types**

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

495       **5.4.2 NamespacePath**

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

497       Conformant WBEM protocols shall support all characteristics of *NamespacePath* values and may support  
498       additional characteristics.

499       **5.4.3 InstancePath**

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

501       Conformant WBEM protocols shall support all characteristics of *InstancePath* values and may support  
502       additional characteristics.

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

505       **5.4.4 ClassPath**

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

507       Conformant WBEM protocols shall support all characteristics of *ClassPath* values and may support  
508       additional characteristics.

509       **5.4.5 QualifierTypePath**

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

511 Conformant WBEM protocols shall support all characteristics of *ClassPath* values may support additional  
512 characteristics.

#### 513 **5.4.6 InstanceSpecification**

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

- 516 • name of the creation class of the instance
- 517 • all or a subset of the static and non-static properties exposed by the creation class of the  
518 instance

519 Each property in an *InstanceSpecification* shall contain:

- 520 • name of the property
- 521 • value of the property
- 522 • optional: Class origin of the property
- 523 • optional: Data type of the property

524 *InstanceSpecification* does not contain the instance path of the CIM instance, because there are some  
525 situations in which the instance data is needed without an instance path. The  
526 *InstanceSpecificationWithPath* type is used when the instance path is needed in addition to the instance  
527 data.

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

#### 529 **5.4.7 ClassSpecification**

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

- 532 • name of the class
- 533 • name of the superclass, if any
- 534 • all or a subset of the static and non-static properties (that is, the property definitions) exposed  
535 by the class. As defined in [DSP0004](#), the set of properties exposed by a class includes any  
536 properties inherited from superclasses, where overridden properties are included only once.
- 537 • all of the static and non-static methods exposed by the class. As defined in [DSP0004](#), the set of  
538 methods exposed by a class includes any methods inherited from superclasses, where  
539 overridden methods are included only once.
- 540 • optional: all of the qualifiers exposed by the class that are defined on the class or any of its  
541 superclasses

542 Each property in a *ClassSpecification* shall contain:

- 543 • name of the property
- 544 • data type of the property
- 545 • default value of the property

- 546       • optional: all of the qualifiers exposed by the property that are defined on the property or any of  
547 its overridden properties

548 Each method in a *ClassSpecification* shall contain:

- 549       • name of the method
- 550       • data type of the return value of the method
- 551       • all of the parameters of the method
- 552       • optional: all of the qualifiers exposed by the method that are defined on the method or any of its  
553 overridden methods

554 Each parameter in that method shall contain:

- 555       • name of the parameter
- 556       • data type of the parameter
- 557       • optional: all of the qualifiers exposed by the parameter that are defined on the parameter or the  
558 corresponding parameter in any of its overridden methods

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

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

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

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

#### 565 **5.4.8 QualifierType**

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

- 568       • name of the qualifier
- 569       • data type of the qualifier
- 570       • default value of the qualifier
- 571       • all flavors of the qualifier
- 572       • all scopes of the qualifier

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

#### 575 **5.4.9 InstanceSpecificationWithPath**

576 A value of the generic type *InstanceSpecificationWithPath* combines the content of *InstanceSpecification*  
577 and *InstancePath*.

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

#### 579 **5.4.10 ClassSpecificationWithPath**

580 A value of the generic type *ClassSpecificationWithPath* combines the content of *ClassSpecification* and  
581 *ClassPath*.

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

#### 583 **5.4.11 QualifierTypeWithPath**

584 A value of the generic type *QualifierTypeWithPath* combines the content of *QualifierType* and  
585 *QualifierTypePath*.

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

#### 587 **5.4.12 ClassName**

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

#### 589 **5.4.13 PropertyName**

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

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

#### 592 **5.4.14 MethodName**

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

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

#### 595 **5.4.15 ParameterValue**

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

- 598 • name of the parameter
- 599 • value of the parameter
- 600 • optional: Data type of the parameter

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

#### 602 **5.4.16 ReturnValue**

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

- 605 • return value
- 606 • optional: Data type of the return value

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

#### 608 **5.4.17 QueryString**

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

#### 611 **5.4.18 QueryLanguage**

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

#### 613 **5.4.19 EnumerationContext**

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

### 616 **5.5 Success and failure**

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

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

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

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

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

### 626 **5.6 Preconditions and postconditions**

627 Each generic operation specifies a set of zero or more preconditions and a set of zero or more  
628 postconditions.

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

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

### 634 **5.7 Generic error messages**

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

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

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

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

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

656 **Mandatory**

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

660 **Conditional**

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

664 **Optional**

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

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

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

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

679 **5.8 Consistency model**

680 This subclause defines consistency requirements for generic operations.

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

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

685 **5.8.1 Definition of ACID properties**

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

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

### 692 **5.8.1.1 Atomicity**

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

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

### 697 **5.8.1.2 Update consistency**

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

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

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

### 703 **5.8.1.3 Isolation**

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

707 Isolation applies to operations and methods that retrieve information through the management interface,  
708 and to operations that modify the managed environment or CIM instances through the management  
709 interface.

### 710 **5.8.1.4 Durability**

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

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

## 716 **5.8.2 Time consistency within a CIM instance**

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

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

### 723 **5.8.3 Staleness of information returned**

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

### 727 **5.8.4 Isolation between operations**

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

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

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

### 738 **5.8.5 Duplicate return of CIM objects or object paths**

739 Any generic operations returning CIM object specifications or CIM object paths should not return  
740 duplicate objects or duplicate object paths.

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

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

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

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

### 752 **5.8.6 Time consistency between returned CIM objects**

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

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 result set may contain instances that represent different points in time.

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

### 760 **5.8.7 Order of returned CIM objects**

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

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

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

### 768 **5.8.8 Validity of returned object paths**

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

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

### 773 5.8.9 Effects of deleting an instance

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

- 807 – by propagating the deletion of the referenced instance to its referencing association  
 808 instance
- 809 – by rejecting the deletion of the referenced instance to be deleted.

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

- 812 Support of the *IfDelete* and *Delete* qualifiers by a WBEM server is optional, as defined in  
813 [DSP0004](#).
- 814 This concept can be used to propagate deletion from an instance to its referencing association  
815 instance, from an association instance to its referenced instances, and in combination also  
816 between associated instances.
- 817 The definition of the *IfDelete* and *Delete* qualifiers in [DSP0004](#) requires that this case is handled  
818 by propagating the deletion of an instance to which the *IfDelete* qualifier applies, to any  
819 instances to which the corresponding *Delete* qualifier applies.
- 820 • **Multiplicity underflow:** Instances associated to an instance to be deleted via an association  
821 with a minimum multiplicity (as defined with *Min* qualifier in the schema, or as constrained by  
822 management profiles) larger than 0 on the reference to the instance to be deleted, if the deletion  
823 would violate the minimum multiplicity that is required.  
824 EXAMPLE: Association AB references class A with *Min* (2) and references class B. Therefore, each  
825 instance of B is supposed to be associated via AB with least two instances of A. If an instance of A is to  
826 be deleted, and there is only one other instance of A associated to the instance of B that is associated  
827 with the instance of A to be deleted, the minimum multiplicity would be violated by the deletion.
- 828 This case shall be handled with any or a combination of the following options:
- 829 – by propagating the deletion of the instance to be deleted to its associated instance defining  
830 the multiplicity constraint, and the association instance.
  - 831 – by rejecting the original deletion.

832 **6 Generic operations**

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

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

Group	Generic Operation
Instance	GetInstance
	DeleteInstance
	ModifyInstance
	CreateInstance
Direct instance enumeration operations	GetClassInstancesWithPath
	GetClassInstancePaths
	GetAssociatedInstancesWithPath
	GetAssociatedInstancePaths
	GetReferencingInstancesWithPath
	GetReferencingInstancePaths
Pulled instance enumeration operations	OpenClassInstancesWithPath
	OpenClassInstancePaths
	OpenAssociatedInstancesWithPath
	OpenAssociatedInstancePaths
	OpenReferencingInstancesWithPath
	OpenReferencingInstancePaths
	OpenQueryInstances
	PullInstancesWithPath
	PullInstancePaths
	PullInstances
	CloseEnumeration
	EnumerationCount
	Method invocation
InvokeStaticMethod	
Class	GetClass
	DeleteClass
	ModifyClass
	CreateClass

Group	Generic Operation
Class enumeration operations	GetTopClassesWithPath
	GetTopClassPaths
	GetSubClassesWithPath
	GetSubClassPaths
	GetAssociatedClassesWithPath
	GetAssociatedClassPaths
	GetReferencingClassesWithPath
	GetReferencingClassPaths
Qualifier type operations	GetQualifierType
	DeleteQualifierType
	ModifyQualifierType
	CreateQualifierType
	EnumerateQualifierTypesWithPath

835 **6.1 Description format**

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

838 **Purpose:**

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

840 **Operation Input Parameters:**

841

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>
...	...	...	...

842 **Operation Output Parameters:**

843

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

844 **Description:**

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

847 **Preconditions:**

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

853 **Postconditions:**

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

857 **Error messages:**

858

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

859

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

862 <diname> Generic name of the operation parameter.

863 <ditype> Generic type of the operation parameter, as defined in 5.4.

864 <direq> Requirement level of the operation parameter, as defined in 5.3.3.

865 <msgid> Message ID of the message, as defined in a DMTF message registry. The message  
866 ID is the concatenation of the values of the XML attributes  
867 MESSAGE/MESSAGE\_ID@PREFIX and  
868 MESSAGE/MESSAGE\_ID@SEQUENCE\_NUMBER.

869 <msgname> Message name of the message, as defined in a DMTF message registry. The  
870 message name is the value of the XML attribute MESSAGE@NAME.

871 <msgreq> Requirement level of the message, as defined in 5.7.

872 <msgsrc> Sources of the message. One or more values may be specified. Valid values are:

873 Infrastructure – the message is implemented by the common infrastructure portion  
874 of the WBEM server.

875 Class implem. – the message is implemented by the class specific portion of the  
876 WBEM server.

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

## 880 **6.2 Common operation parameters for all operations**

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

### 884 **6.2.1 IncludeClassOrigin**

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

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

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

892 If the WBEM protocol supports client side control of returning class origin information, then the  
893 *IncludeClassOrigin* operation parameter shall be supported. If the *IncludeClassOrigin* operation  
894 parameter is TRUE, then class origin information shall be included for any element in any object returned  
895 by the operation. If the *IncludeClassOrigin* operation parameter is FALSE, then class origin information  
896 shall not be included for any element in any object returned by the operation.

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

### 899 **6.2.2 IncludeQualifiers**

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

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

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

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

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

### 911 **6.2.3 <element>List**

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

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

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

919 **6.3 Instance operations**

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

922 **6.3.1 GetInstance**

923 **Purpose:**

924 Retrieves a CIM instance.

925 **Operation Input Parameters:**

926

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

927 **Operation Output Parameters:**

928

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

929 **Description:**

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

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

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

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

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

948   **Preconditions:**

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

951   **Postconditions:**

- 952                   • The instance shall have been returned with the properties as defined in the Description  
953                   paragraph for this operation.
- 954                   • Requirements on ACID properties:
- 955                   – Atomicity: N/A
- 956                   – Update Consistency: N/A
- 957                   – Isolation: Required
- 958                   – Durability: N/A

959   **Error messages:**

960

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.	

961 **6.3.2 DeleteInstance**

962 **Purpose:**

963 Deletes a CIM instance.

964 **Operation Input Parameters:**

965

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

966 **Operation Output Parameters:**

967 None.

968 **Description:**

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

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

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

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

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

983 **Preconditions:**

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

986 **Postconditions:**

- 987 • The instance referenced by *InstancePath* shall have been deleted.
- 988 • Any implicit deletions of dependent CIM instances shall have happened, as defined in 5.8.9.
- 989 • Any effects of the deletion of all of these CIM instances on any underlying resources shall have  
990 happened.
- 991 • The consistency requirements defined in [DSP0004](#) shall be satisfied for any instances related to  
992 the deleted instances.
- 993 • Requirements on ACID properties:
  - 994 – Atomicity: Required, if dependent instances are handled by rejection, as defined in 5.8.9.
  - 995 Recommended, if dependent instances are handled by delete propagation, as defined in  
996 5.8.9.

- 997           – Update Consistency: Required, if dependent instances are handled by rejection, as defined  
998           in 5.8.9. Recommended, if dependent instances are handled by delete propagation, as  
999           defined in 5.8.9.
- 1000          – Isolation: Required, if dependent instances are handled by rejection, as defined in 5.8.9.  
1001          Recommended, if dependent instances are handled by delete propagation, as defined in  
1002          5.8.9.
- 1003          – Durability: Required.

1004       **Error messages:**

1005

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.	

1006 **6.3.3 ModifyInstance**

1007 **Purpose:**

1008 Changes property values of a CIM instance.

1009 **Operation Input Parameters:**

1010

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

1011 **Operation Output Parameters:**

1012 None.

1013 **Description:**

1014 The *ModifyInstance* operation changes property values of the instance referenced by *InstancePath*.

1015 The set of properties to be changed shall be determined using the following algorithm:

- 1016 • Initially, the set of properties to be changed is the set of properties specified in  
1017 *ModifiedInstance*.
- 1018 • If the *IncludedProperties* operation input parameter is supported by the WBEM protocol  
1019 and if its value is not NULL, it acts as a restricting filter on the properties to be changed  
1020 such that any properties exposed by the creation class of the instance that are not named  
1021 in that operation parameter are removed from the set of properties to be changed. Any  
1022 duplicate or invalid property names in the *IncludedProperties* operation input parameter  
1023 shall be ignored. A non-NULL empty *IncludedProperties* list removes all properties from  
1024 that set.
- 1025 • Any key properties and non-modifiable properties are removed from the set of properties to  
1026 be changed. As a result, specifying such properties in *ModifiedInstance* or  
1027 *IncludedProperties* does not cause an error.

1028 NOTE: The modifiability of properties can be defined in the schema and in management profiles.

1029 Conformant WBEM protocols may restrict *ModifiedInstance* to specify all properties exposed by the  
1030 creation class of the instance referenced by *InstancePath*.

1031 **Preconditions:**

- 1032 • The instance referenced by *InstancePath* shall exist in the namespace. If this is not satisfied,  
1033 the operation shall fail, indicating WIPG0213.
- 1034 • The creation class of *ModifiedInstance* shall be the creation class of the instance referenced by  
1035 *InstancePath* or a superclass of that class. If this is not satisfied, the operation shall fail,  
1036 indicating WIPG0208.

- 1037       • Any properties specified in *ModifiedInstance* shall be from the set of properties exposed by the  
 1038       creation class of *ModifiedInstance*. If this is not satisfied, the operation shall fail, indicating  
 1039       WIPG0208.

1040       **Postconditions:**

- 1041       • The values of the properties shall have been modified as defined in the Description paragraph  
 1042       for this operation.
- 1043       • The values of key properties and non-modifiable properties shall not have been modified.
- 1044       • Other properties may have changed as a result of side effects of changing properties, behavior  
 1045       defined in referencing specifications, or volatility of properties.
- 1046       • The consistency requirements defined in [DSP0004](#) shall be satisfied for the modified instance.
- 1047       • Requirements on ACID properties:
- 1048       – Atomicity: Required
- 1049       – Update Consistency: Required
- 1050       – Isolation: Required
- 1051       – Durability: Required

1052       **Error messages:**

1053

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.	

1054 **6.3.4 CreateInstance**

1055 **Purpose:**

1056 Creates a CIM instance.

1057 **Operation Input Parameters:**

1058

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

1059 **Operation Output Parameters:**

1060

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

1061 **Description:**

1062 The *CreateInstance* operation creates a CIM instance in the namespace specified in *ClassPath* from  
 1063 the creation class specified in *ClassPath*, and returns the instance path of the new instance.

1064 As defined in the description of the *ClassPath* type, the class path of the CIM class to be used as a  
 1065 creation class for the instance is interpreted in a non-polymorphic way, i.e., it references the  
 1066 specified class only and not any subclasses. In other words, the instance is created from the  
 1067 specified class only. As a result, the specified class becomes the creation class of the instance.

1068 The newly created instance shall have all properties exposed by the creation class specified in  
 1069 *ClassPath*.

1070 For each property, the initial value shall be determined as defined in the following **default** rules:

- 1071 • If the *NewInstance* operation input parameter is supported, and if the property is included  
 1072 in *NewInstance*, its value is used. That is also the case if that value is NULL.
- 1073 • Else, if a default value is declared for the property, that value is used.

1074 These default rules allow specifying key properties and non-writeable properties in *NewInstance*. In  
 1075 other words, the creation of an instance does not have the restrictions a subsequent modification  
 1076 has.

1077 As defined in [DSP1001](#), management profiles may specify any such rules, overriding these default  
 1078 rules. This may result in rejecting, respecting or replacing the values of any properties specified in  
 1079 *NewInstance*, as well as respecting or replacing the default values of any properties not specified in  
 1080 *NewInstance*.

1081 Volatile properties may change their values immediately after the instance has been created.

1082 Instance creation based upon input data other than initial property values can be done using CIM  
 1083 methods. For example, creation of an instance of *CIM\_ComputerSystem* representing a virtual  
 1084 computer system could be done using a *CreateVirtualComputerSystem( )* method taking a higher-  
 1085 level specification of the virtual computer system as input.

1086 Other CIM instances may come into existence implicitly during the course of processing the  
 1087 *CreateInstance* operation. As defined in [DSP1001](#), management profiles may specify the rules for  
 1088 such implicitly created instances.

1089 Any such implicitly created instances may reside in a different CIM namespace (which may reside in  
 1090 a different WBEM server) than the namespace specified in *ClassPath*.

1091 In case of error, the consistency requirements defined in [DSP0004](#) should be attempted to be  
 1092 satisfied in a best effort approach. In case of error, only a subset of the instances to be created may  
 1093 have been created, but each instance shall have either been created completely or not at all.

1094 As defined in [DSP1001](#), management profiles may specify the effects of the creation of CIM  
 1095 instances on their underlying resources. For example, a management profile may define that the  
 1096 lifecycle of the CIM instance is coupled with the lifecycle of some underlying resource, and that this  
 1097 resource shall be created when the instance is created.

#### 1098 **Preconditions:**

- 1099 • The instance to be created shall not exist in the namespace specified by *ClassPath*. If this is not  
 1100 satisfied, the operation shall fail, indicating WIPG0216.
- 1101 • The CIM class referenced by *ClassPath* shall exist in the namespace. If this is not satisfied, the  
 1102 operation shall fail, indicating WIPG0214.
- 1103 • The creation class of *NewInstance* shall be the class referenced by *ClassPath* or a superclass  
 1104 of that class. If this is not satisfied, the operation shall fail, indicating WIPG0208.
- 1105 • Any properties specified in *NewInstance* shall be from the set of properties exposed by the  
 1106 class referenced by *ClassPath*. If this is not satisfied, the operation shall fail, indicating  
 1107 WIPG0208.

#### 1108 **Postconditions:**

- 1109 • The instance shall have been created as defined in the Description paragraph for this operation.
- 1110 • Any management profile defined implicit creations of other CIM instances shall have happened.
- 1111 • Any management profile defined effects of the creation of all of these CIM instances on any  
 1112 underlying resources shall have happened.
- 1113 • Requirements on ACID properties:
  - 1114 – Atomicity: Required
  - 1115 – Update Consistency: Required
  - 1116 – Isolation: Required
  - 1117 – Durability: Required

#### 1118 **Error messages:**

1119

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.	

## 1120 6.4 Direct instance enumeration operations

1121 This subclause defines direct instance enumeration operations (operations that enumerate CIM instances  
1122 and return them directly as a result of the operation).

### 1123 6.4.1 GetClassInstancesWithPath

#### 1124 Purpose:

1125 Enumerate the CIM instances of a class and return these instances.

#### 1126 Operation Input Parameters:

1127

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

#### 1128 Operation Output Parameters:

1129

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

#### 1130 Description:

1131 The *GetClassInstancesWithPath* operation enumerates all CIM instances of the class referenced by  
1132 *EnumClassPath*, including instances of any of its subclasses, and returns these instances together  
1133 with their instance paths.

1134 All of the instances returned shall exist in the same namespace as the class referenced by  
1135 *EnumClassPath*.

1136 An instance is included in the result set if and only if it exists in the namespace specified in  
1137 *EnumClassPath*, and its creation class is the class specified in *EnumClassPath* or a subclass of that  
1138 class.

1139 The result set should not contain any duplicate instances, as defined in 5.8.4. Because the result set  
1140 contains only instances that exist in the same namespace, a determination of duplicate instances (for  
1141 example by the Client) can be done on the basis of their model paths only.

- 1142 The set of properties to be included in any instances in the result set shall be determined using the  
 1143 following algorithm:
- 1144 • Initially, the set of properties to be included is the set of properties exposed by the creation  
 1145 class of the instance. This includes all the duplicates of any duplicate non-overridden  
 1146 properties.
  - 1147 • If the *IncludedProperties* operation input parameter is supported by the WBEM protocol  
 1148 and if its value is not NULL, it acts as a restricting filter on the properties to be included in  
 1149 the returned instances such that any properties exposed by the creation class of the  
 1150 instance that are not named in that operation parameter are removed from the set of  
 1151 properties to be included. Any duplicate or invalid property names in the  
 1152 *IncludedProperties* operation input parameter shall be ignored. A non-NULL empty  
 1153 *IncludedProperties* list removes all properties from the set of properties to be included.
  - 1154 • If the *ExcludeSubclassProperties* operation input parameter is supported by the WBEM  
 1155 protocol and if its value is TRUE, it acts as a restricting filter on the properties to be  
 1156 included in the returned instances such that any properties not exposed by the class  
 1157 referenced by *EnumClassPath* are removed from the set of properties to be included. In  
 1158 other words, the set of properties is restricted to the properties exposed by the  
 1159 enumeration class.
  - 1160 • Conformant WBEM protocols may specify rules that cause properties with a value of NULL  
 1161 to be removed from the set of properties to be included.

1162 **Preconditions:**

- 1163 • The CIM class referenced by *EnumClassPath* shall exist in the namespace. If this is not  
 1164 satisfied, the operation shall fail, indicating WIPG0214.

1165 **Postconditions:**

- 1166 • The enumerated instances with their instance paths shall have been returned as described in  
 1167 the Description paragraph for this operation.
- 1168 • Requirements on ACID properties:
  - 1169 – Atomicity: N/A
  - 1170 – Update Consistency: N/A
  - 1171 – Isolation: Required at the level of single instances, as defined in 5.8.
  - 1172 – Durability: N/A

1173 **Error Messages:**  
 1174

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.	

## 1175 6.4.2 GetClassInstancePaths

### 1176 Purpose:

1177 Enumerate the CIM instances of a class and return their instance paths.

### 1178 Operation Input Parameters:

1179

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

### 1180 Operation Output Parameters:

1181

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

### 1182 Description:

1183 The *GetClassInstancePaths* operation enumerates all CIM instances of the class referenced by  
1184 *EnumClassPath*, and returns the instance paths of these instances.

1185 An instance is included in the result set if and only if it exists in the namespace specified in  
1186 *EnumClassPath*, and its creation class is the class specified in *EnumClassPath* or a subclass of that  
1187 class.

1188 The result set should not contain any duplicate instances, as defined in 5.8.4. Because the result set  
1189 contains only instances that exist in the same namespace, a determination of duplicate instances (for  
1190 example by the Client) can be done on the basis of their model paths only.

1191 **Preconditions:**

- 1192 • The CIM class referenced by *EnumClassPath* shall exist in the namespace. If this is not
- 1193 satisfied, the operation shall fail, indicating WIPG0214.

1194 **Postconditions:**

- 1195 • The instance paths of the enumerated instances shall have been returned as described in the
- 1196 Description paragraph for this operation.
- 1197 • Requirements on ACID properties:
  - 1198 – Atomicity: N/A
  - 1199 – Update Consistency: N/A
  - 1200 – Isolation: Required at the level of single instances, as defined in 5.8.
  - 1201 – Durability: N/A

1202 **Error messages:**

1203

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.	

1204 **6.4.3 GetAssociatedInstancesWithPath**1205 **Purpose:**

1206 Enumerate CIM instances that are associated with a given source instance and return those  
 1207 instances together with their instance paths.

1208 **Operation Input Parameters:**  
1209

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

1210 **Operation Output Parameters:**  
1211

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

1212 **Description:**

1213 The *GetAssociatedInstancesWithPath* operation enumerates instances that are associated with a  
 1214 given source instance and returns these instances together with their instance paths.

1215 The set of associated instances to be returned shall be determined using the following algorithm:

- 1216 • Initially, the set of instances to be returned is the set of all instances associated to the
- 1217 source instance specified in *SourceInstancePath*. The associations may be instances of
- 1218 different association classes.

1219 The result set should not contain any duplicate instances, as defined in 5.8.4. However,

1220 different far ends may reference the same instance, and in such cases, the instance shall

1221 be contained in the result set once for each such reference.

- 1222 • If the *AssociationClassName* operation input parameter is not NULL, it acts as a restricting
- 1223 filter on the instances to be returned such that each instance that is associated with the
- 1224 source instance using an association whose creation class or one of its superclasses does
- 1225 not have the name specified in *AssociationClassName*, is removed from the set of
- 1226 instances to be returned. There shall be no validity checking performed for the
- 1227 *AssociationClassName* operation input parameter.

- 1228 • If the *AssociatedClassName* operation input parameter is not NULL, it acts as a restricting
- 1229 filter on the instances to be returned such that each instance whose creation class or one
- 1230 of its superclasses does not have the name specified in *AssociatedClassName*, is removed
- 1231 from the set of instances to be returned. There shall be no validity checking performed for
- 1232 the *AssociatedClassName* operation input parameter.

- 1233 • If the *SourceRoleName* operation input parameter is not NULL, it acts as a restricting filter
- 1234 on the instances to be returned such that each instance that is associated with the source
- 1235 instance using an association class that has a role name on the source end that is not the
- 1236 role name specified in *SourceRoleName*, is removed from the set of instances to be
- 1237 returned. There shall be no validity checking performed for the *SourceRoleName* operation
- 1238 input parameter.

- 1239 • If the *AssociatedRoleName* operation input parameter is not NULL, it acts as a restricting
- 1240 filter on the instances to be returned such that each instance that is associated with the
- 1241 source instance using an association class that has a role name on the end referencing
- 1242 that instance that is not the role name specified in *AssociatedRoleName*, is removed from
- 1243 the set of instances to be returned. There shall be no validity checking performed for the
- 1244 *AssociatedRoleName* operation input parameter.

1245 The set of properties to be included in each returned associated instance shall be determined using

1246 the following algorithm:

- 1247 • Initially, the set of properties to be included is the set of properties exposed by the creation
- 1248 class of the instance. This includes all the duplicates of any duplicate non-overridden
- 1249 properties.

- 1250 • If the *IncludedProperties* operation input parameter is supported by the WBEM protocol
- 1251 and if its value is not NULL, it acts as a restricting filter on the properties to be included in
- 1252 the returned instances such that any properties exposed by the creation class of the
- 1253 instance that are not named in that operation parameter are removed from the set of
- 1254 properties to be included. Any duplicate or invalid property names in the
- 1255 *IncludedProperties* operation input parameter shall be ignored. A non-NULL empty
- 1256 *IncludedProperties* list removes all properties from the set of properties to be included.

- 1257 • If the *ExcludeSubclassProperties* operation input parameter is supported by the WBEM
- 1258 protocol and if its value is TRUE, it acts as a restricting filter on the properties to be
- 1259 included in the returned instances such that any properties not exposed by the class
- 1260 specified in *AssociatedClassName* are removed from the set of properties to be included.

- 1261 • Conformant WBEM protocols may specify rules that cause properties with a value of NULL
- 1262 to be removed from the set of properties to be included.

1263 **Preconditions:**

- 1264 • The instance referenced by *SourceInstancePath* shall exist in the namespace. If this is not  
1265 satisfied, the operation shall fail, indicating WIPG0213.
- 1266 • The *IncludedProperties* operation parameter, if supported by the WBEM protocol, shall only be  
1267 specified with a non-NULL value if the *AssociatedClassName* operation input parameter is also  
1268 non-NULL. If this is not satisfied, the operation shall fail, indicating WIPG0208.
- 1269 • The *ExcludeSubclassProperties* operation parameter, if supported by the WBEM protocol, shall  
1270 only be specified with a TRUE value if the *AssociatedClassName* operation input parameter is  
1271 non-NULL. If this is not satisfied, the operation shall fail, indicating WIPG0208.

1272 NOTE: Specifying a non-NULL value for *AssociatedClassName* ensures that the associated instances have the  
1273 class specified in *AssociatedClassName* as a common superclass.

1274 **Postconditions:**

- 1275 • The associated instances with their instance paths shall have been returned as described in the  
1276 Description paragraph for this operation.
- 1277 • Requirements on ACID properties:
  - 1278 – Atomicity: N/A
  - 1279 – Update Consistency: N/A
  - 1280 – Isolation: Required at the level of single instances, as defined in 5.8.
  - 1281 – Durability: N/A

1282 **Error Messages:**

1283

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.	

1284 **6.4.4 GetAssociatedInstancePaths**

1285 **Purpose:**

1286 Enumerate CIM instances that are associated with a given source instance and return their instance  
 1287 paths.

1288 **Operation Input Parameters:**

1289

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

1290 **Operation Output Parameters:**

1291

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

1292 **Description:**

1293 The *GetAssociatedInstancePaths* operation enumerates the instance paths of instances that are  
 1294 associated with a given source instance and returns these instance paths.

1295 The set of associated instances of which instance paths are to be returned shall be determined using  
 1296 the following algorithm:

- 1297 • Initially, the set of instances to be returned is the set of all instances associated to the  
 1298 source instance specified in *SourceInstancePath*. The associations may be instances of  
 1299 different association classes.

1300 The result set should not contain any duplicate instances, as defined in 5.8.4. However,  
 1301 different association instances may reference the same instance on one of their far ends,

- 1302 and in such cases, the instance shall be contained in the result set once for each such  
1303 reference.
- 1304 • If the *AssociationClassName* operation input parameter is not NULL, it acts as a restricting  
1305 filter on the instances to be returned such that each instance that is associated with the  
1306 source instance using an association whose creation class or one of its superclasses does  
1307 not have the name specified in *AssociationClassName*, is removed from the set of  
1308 instances to be returned. There shall be no validity checking performed for the  
1309 *AssociationClassName* operation input parameter.
  - 1310 • If the *AssociatedClassName* operation input parameter is not NULL, it acts as a restricting  
1311 filter on the instances to be returned such that each instance whose creation class or one  
1312 of its superclasses does not have the name specified in *AssociatedClassName*, is removed  
1313 from the set of instances to be returned. There shall be no validity checking performed for  
1314 the *AssociatedClassName* operation input parameter.
  - 1315 • If the *SourceRoleName* operation input parameter is not NULL, it acts as a restricting filter  
1316 on the instances to be returned such that each instance that is associated with the source  
1317 instance using an association class that has a role name on the source end that is not the  
1318 role name specified in *SourceRoleName*, is removed from the set of instances to be  
1319 returned. There shall be no validity checking performed for the *SourceRoleName* operation  
1320 input parameter.
  - 1321 • If the *AssociatedRoleName* operation input parameter is not NULL, it acts as a restricting  
1322 filter on the instances to be returned such that each instance that is associated with the  
1323 source instance using an association class that has a role name on the end referencing  
1324 that instance that is not the role name specified in *AssociatedRoleName*, is removed from  
1325 the set of instances to be returned. There shall be no validity checking performed for the  
1326 *AssociatedRoleName* operation input parameter.

1327 The consistency model defined in 5.8 applies.

1328 **Preconditions:**

- 1329 • The instance referenced by *SourceInstancePath* shall exist in the namespace. If this is not  
1330 satisfied, the operation shall fail, indicating WIPG0213.

1331 **Postconditions:**

- 1332 • The instance paths of the associated instances shall have been returned as described in the  
1333 Description paragraph for this operation.
- 1334 • Requirements on ACID properties:
  - 1335 – Atomicity: N/A
  - 1336 – Update Consistency: N/A
  - 1337 – Isolation: Required at the level of single instances, as defined in 5.8.
  - 1338 – Durability: N/A

1339 **Error Messages:**  
1340

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.	

1341 **6.4.5 GetReferencingInstancesWithPath**1342 **Purpose:**

1343 Enumerate CIM association instances that reference a given source instance and return these  
 1344 instances together with their instance path.

1345 **Operation Input Parameters:**

1346

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

1347 **Operation Output Parameters:**

1348

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

1349 **Description:**

1350 The *GetReferencingInstancesWithPath* operation enumerates association instances that reference  
 1351 the specified source instance and returns these instances together with their instance paths.

- 1352 The set of association instances to be returned shall be determined using the following algorithm:
- 1353 • Initially, the set of instances to be returned is the set of all instances referencing the source  
1354 instance specified in *SourceInstancePath*. These associations may be instances of  
1355 different association classes.
  - 1356 • If the *AssociationClassName* operation input parameter is not NULL, it acts as a restricting  
1357 filter on the instances to be returned such that each association instance whose creation  
1358 class or one of its superclasses does not have the name specified in  
1359 *AssociationClassName*, is removed from the set of instances to be returned. There shall be  
1360 no validity checking performed for the *AssociationClassName* operation input parameter.
  - 1361 • If the *AssociatedClassName* operation input parameter is not NULL, it acts as a restricting  
1362 filter on the instances to be returned such that each association instance whose creation  
1363 class has a set of far ends none of which is referencing a class where that class or one of  
1364 its superclasses has the name specified in *AssociatedClassName*, is removed from the set  
1365 of instances to be returned. There shall be no validity checking performed for the  
1366 *AssociatedClassName* operation input parameter.
  - 1367 • If the *SourceRoleName* operation input parameter is not NULL, it acts as a restricting filter  
1368 on the instances to be returned such that each association instance whose creation class  
1369 does not have the role name specified in *SourceRoleName* on the end referencing the  
1370 source instance, is removed from the set of instances to be returned. There shall be no  
1371 validity checking performed for the *SourceRoleName* operation input parameter.
  - 1372 • If the *AssociatedRoleName* operation input parameter is not NULL, it acts as a restricting  
1373 filter on the instances to be returned such that each association instance whose creation  
1374 class has a set of far ends none of which has the role name specified in  
1375 *AssociatedRoleName*, is removed from the set of instances to be returned. There shall be  
1376 no validity checking performed for the *AssociatedRoleName* operation input parameter.
- 1377 The consistency model defined in 5.8 applies.
- 1378 The set of properties to be included in each returned association instance shall be determined using  
1379 the following algorithm:
- 1380 • Initially, the set of properties to be included is the set of properties exposed by the creation  
1381 class of the instance. This includes all the duplicates of any duplicate non-overridden  
1382 properties.
  - 1383 • If the *IncludedProperties* operation input parameter is supported by the WBEM protocol  
1384 and if its value is not NULL, it acts as a restricting filter on the properties to be included in  
1385 the returned instances such that any properties exposed by the creation class of the  
1386 instance that are not named in that operation parameter are removed from the set of  
1387 properties to be included. Any duplicate or invalid property names in the  
1388 *IncludedProperties* operation input parameter shall be ignored. A non-NULL empty  
1389 *IncludedProperties* list removes all properties from the set of properties to be included
  - 1390 • If the *ExcludeSubclassProperties* operation input parameter is supported by the WBEM  
1391 protocol and if its value is TRUE, it acts as a restricting filter on the properties to be  
1392 included in the returned instances such that any properties not exposed by the class  
1393 specified in *AssociationClassName* are removed from the set of properties to be included.
  - 1394 • Conformant WBEM protocols may specify rules that cause properties with a value of NULL  
1395 to be removed from the set of properties to be included.

1396 **Preconditions:**

- 1397 • The instance referenced by *SourceInstancePath* shall exist in the namespace. If this is not  
1398 satisfied, the operation shall fail, indicating WIPG0213.
- 1399 • The *IncludedProperties* operation parameter, if supported by the WBEM protocol, shall only be  
1400 specified with a non-NULL value if the *AssociationClassName* operation input parameter is also  
1401 non-NULL. If this is not satisfied, the operation shall fail, indicating WIPG0208.
- 1402 • The *ExcludeSubclassProperties* operation parameter, if supported by the WBEM protocol, shall  
1403 only be specified with a TRUE value if the *AssociationClassName* operation input parameter is  
1404 non-NULL. If this is not satisfied, the operation shall fail, indicating WIPG0208.

1405 NOTE: Specifying a non-NULL value for *AssociationClassName* ensures that the association instances have the  
1406 class specified in *AssociationClassName* as a common superclass.

1407 **Postconditions:**

- 1408 • The association instances with their instance paths shall have been returned as described in the  
1409 Description paragraph for this operation.
- 1410 • Requirements on ACID properties:
  - 1411 – Atomicity: N/A
  - 1412 – Update Consistency: N/A
  - 1413 – Isolation: Required at the level of single instances, as defined in 5.8.
  - 1414 – Durability: N/A

1415 **Error Messages:**

1416

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.	

1417 **6.4.6 GetReferencingInstancePaths**

1418 **Purpose:**

1419 Enumerate CIM association instances that reference a given source instance and return their  
 1420 instance paths.

1421 **Operation Input Parameters:**

1422

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

1423 **Operation Output Parameters:**

1424

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

1425 **Description:**

1426 The *GetReferencingInstancePaths* operation enumerates the instance paths of association instances  
 1427 that reference the specified source instance and returns these instance paths.

1428 The set of association instances of which instance paths are to be returned shall be determined  
 1429 using the following algorithm:

- 1430 • Initially, the set of instances to be returned is the set of all instances referencing the source  
 1431 instance specified in *SourceInstancePath*. These associations may be instances of  
 1432 different association classes.
- 1433 • If the *AssociationClassName* operation input parameter is not NULL, it acts as a restricting  
 1434 filter on the instances to be returned such that each association instance whose creation

- 1435 class or one of its superclasses does not have the name specified in  
 1436 *AssociationClassName*, is removed from the set of instances to be returned. There shall be  
 1437 no validity checking performed for the *AssociationClassName* operation input parameter.
- 1438 • If the *AssociatedClassName* operation input parameter is not NULL, it acts as a restricting  
 1439 filter on the instances to be returned such that each association instance whose creation  
 1440 class has a set of far ends none of which is referencing a class where that class or one of  
 1441 its superclasses has the name specified in *AssociatedClassName*, is removed from the set  
 1442 of instances to be returned. There shall be no validity checking performed for the  
 1443 *AssociatedClassName* operation input parameter.
  - 1444 • If the *SourceRoleName* operation input parameter is not NULL, it acts as a restricting filter  
 1445 on the instances to be returned such that each association instance whose creation class  
 1446 does not have the role name specified in *SourceRoleName* on the end referencing the  
 1447 source instance, is removed from the set of instances to be returned. There shall be no  
 1448 validity checking performed for the *SourceRoleName* operation input parameter.
  - 1449 • If the *AssociatedRoleName* operation input parameter is not NULL, it acts as a restricting  
 1450 filter on the instances to be returned such that each association instance whose creation  
 1451 class has a set of far ends none of which has the role name specified in  
 1452 *AssociatedRoleName*, is removed from the set of instances to be returned. There shall be  
 1453 no validity checking performed for the *AssociatedRoleName* operation input parameter.
- 1454 The consistency model defined in 5.8 applies.

1455 **Preconditions:**

- 1456 • The instance referenced by *SourceInstancePath* shall exist in the namespace. If this is not  
 1457 satisfied, the operation shall fail, indicating WIPG0213.

1458 **Postconditions:**

- 1459 • The instance paths of the association instances shall have been returned as described in the  
 1460 Description paragraph for this operation.
- 1461 • Requirements on ACID properties:
  - 1462 – Atomicity: N/A
  - 1463 – Update Consistency: N/A
  - 1464 – Isolation: Required at the level of single instances, as defined in 5.8.
  - 1465 – Durability: N/A

1466 **Error Messages:**

1467

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.	

## 1468 **6.5 Pulled instance enumeration operations**

1469 This subclause defines pulled instance enumeration operations (operations that enumerate CIM  
1470 instances and return them by means of subsequent pull operations).

1471 The common pattern for these operations is that an enumeration session gets established through an  
1472 "Open" operation, also establishing the kind of operation and the kind of items to be returned (instances  
1473 or instance paths of instances), and subsequent repeated executions of a "Pull" operation on the  
1474 enumeration session are used to retrieve the items. Optionally, the "Open" operation can also pull a first  
1475 set of items.

1476 The pulled instance enumeration operations consist of the following individual operations:

1477 • Open operations:

1478 OpenClassInstancesWithPath – Open an enumeration of instances of a class

1479 OpenClassInstancePaths – Open an enumeration of the instance paths of instances of a class

1480 OpenAssociatedInstancesWithPath – Open an enumeration of instances associated to a source  
1481 instance

1482 OpenAssociatedInstancePaths – Open an enumeration of the instance paths of instances  
1483 associated to a source instance

1484 OpenReferencingInstancesWithPath – Open an enumeration of association instances  
1485 referencing a source instance

1486 OpenReferencingInstancePaths – Open an enumeration of the instance paths of association  
1487 instances referencing a source instance

1488 OpenQueryInstances – Open an enumeration of instances representing a query result

1489 • Pull operations:

1490 PullInstances – Pull operation for retrieving instances with paths

1491 PullInstancePaths – Pull operation for retrieving instance paths

1492 PullInstancesWithoutPath – Pull operation for retrieving instances without paths

1493 • Other operations:

1494 CloseEnumeration – Close an open enumeration

1495 EnumerationCount – Estimate number of items in an open enumeration

### 1496 **6.5.1 General behavioral rules**

1497 A central concept of the pulled instance enumeration operations is the "enumeration session". An  
1498 enumeration session can be thought of as a context in which the operations perform their work, and  
1499 which determines the set of objects to be returned. In order to process the operations related to an  
1500 enumeration session, some of the operation parameters of the Open operation need to be maintained as  
1501 long as the enumeration session is open, as well as some state data about where the enumeration  
1502 session is with respect to objects already returned.

1503 From a WBEM client's perspective, an enumeration session is represented as an enumeration context  
1504 value. A successful Open operation establishes the enumeration session and returns an enumeration  
1505 context value representing the open enumeration session. The enumeration context value is used as an  
1506 operation input/output parameter in subsequent Pull operations on that enumeration session. The

1507 enumeration context value shall uniquely identify the open enumeration session within the target CIM  
 1508 namespace of the Open operation that established the enumeration session. This does not require the  
 1509 enumeration context value to be time-unique, i.e., it may be reused for a new enumeration session after  
 1510 the old enumeration session was closed. It is valid for a WBEM server to use NULL as an enumeration  
 1511 context value representing a closed enumeration session, but a WBEM client shall not rely on that to  
 1512 detect that an enumeration session has been closed.

1513 Defining the enumeration context value in Pull operations not only as an operation input parameter but  
 1514 also as an operation output parameter allows the WBEM server to change the enumeration context value  
 1515 during the execution of a Pull operation. This allows for different implementation approaches for the  
 1516 WBEM server, which are transparent for the WBEM client.

1517 Example approaches are:

- 1518 • maintaining any state data describing the enumeration session internally in the WBEM server.  
 1519 In this approach, the enumeration context value does not need to change in subsequent Pull  
 1520 operations. It is used by the WBEM server only to identify the internal state data for the open  
 1521 enumeration session, but it is not used to store any of the state data in it. A variation of this  
 1522 approach is to hand back modified enumeration context values for additional WBEM server side  
 1523 sequence checking.
- 1524 • maintaining any state data describing the enumeration session on the WBEM client side only. In  
 1525 this approach, all state data is stored in the enumeration context value, and the WBEM server  
 1526 does not maintain any state data about the enumeration session, essentially being completely  
 1527 stateless with respect to the enumeration session.
- 1528 • a combination of the two previous approaches

1529 A WBEM server may support keeping enumeration sessions open across connection terminations and  
 1530 shutdowns of the server. Objects may be created, deleted or modified concurrently with an enumeration  
 1531 session that involves these objects. Such changes may or may not be reflected in the enumeration set.  
 1532 Therefore, there is no guarantee to the WBEM client that the enumeration set represents a consistent  
 1533 snapshot of its objects at a point in time. However, the WBEM server should make a best effort attempt  
 1534 for the returned enumeration set to represent a consistent snapshot of its objects at a point in time. The  
 1535 order of objects in the enumeration set is undefined.

1536 This specification does not define any restrictions on the number of enumeration sessions that can be  
 1537 established or executed on concurrently in the same WBEM server or by the same WBEM client. This  
 1538 remains true even if the enumeration sets of such concurrently established enumeration sessions contain  
 1539 the same objects.

1540 With the exception of CloseEnumeration, all operations on a particular enumeration session shall be  
 1541 executed sequentially. An enumeration session can be open or closed. The enumeration session is  
 1542 considered open if operations using its enumeration context value as an operation input parameter can  
 1543 be executed successfully. It is opened by the successful completion of an Open operation and closed by  
 1544 one of the following:

- 1545 • Successful completion of a CloseEnumeration operation
- 1546 • Successful completion of an Open or Pull operation that has its *EndOfSequence* operation  
 1547 output parameter set to TRUE. In other words, reaching the end of the enumeration set closes  
 1548 the enumeration session implicitly
- 1549 • Unsuccessful completion of a Pull operation when *ContinueOnError* had not been requested
- 1550 • WBEM server side decision to close the enumeration session based upon an operation timeout
- 1551 • WBEM server side decision to close an enumeration session during an operation on that  
 1552 enumeration session based upon exceeding server limits.

1553 A conformant WBEM server may support closure of enumeration sessions based upon exceeding server  
 1554 limits. Potential examples for such a decision may be Pull operations with no objects requested that are  
 1555 repeated with a high frequency on the same enumeration session, or EnumerationCount operations  
 1556 repeated with a high frequency on the same enumeration session. If a WBEM server supports closure of  
 1557 enumeration sessions based upon exceeding server limits, it shall make the decision to close an  
 1558 enumeration session during an operation on that enumeration session. (There is no way to indicate the  
 1559 reason for the closure if the decision is made elsewhere.)

## 1560 **6.5.2 Common operation parameters for the open operations**

1561 This subclause defines commonly used operation parameters for the Open operations. The description of  
 1562 the individual Open operations references these operation parameters as appropriate. However, not  
 1563 every Open operation uses every one of these common operation parameters.

### 1564 **6.5.2.1 EnumerationContext**

1565 The *EnumerationContext* operation output parameter is the enumeration context value representing the  
 1566 enumeration session. See 6.5.1 for a definition of the concepts of *enumeration session* and *enumeration*  
 1567 *context value*.

### 1568 **6.5.2.2 EndOfSequence**

1569 NOTE: This operation output parameter is also used for Pull operations.

1570 The *EndOfSequence* operation output parameter indicates whether the enumeration session is  
 1571 exhausted.

1572 If *EndOfSequence* is TRUE upon successful completion of an operation, no more objects are available  
 1573 and the WBEM server shall have closed the enumeration session, releasing any possibly allocated  
 1574 resources related to the enumeration session.

1575 If the returned enumeration set is empty, it is valid for a WBEM server to set *EndOfSequence* to TRUE,  
 1576 even if *MaxObjectCount* was 0. In this case, the enumeration session will be closed upon successful  
 1577 completion of the operation.

1578 If *EndOfSequence* is FALSE upon successful completion of an operation, there may be additional  
 1579 elements available and the WBEM server shall not have closed the enumeration session.

### 1580 **6.5.2.3 FilterQueryLanguage and FilterQueryString**

1581 The *FilterQueryLanguage* and *FilterQueryString* operation input parameters define a filter query that acts  
 1582 as an additional restricting filter on the set of instances about which information is returned (that is, the  
 1583 instances themselves or their instance paths).

1584 Support for the *FilterQueryLanguage* and *FilterQueryString* operation parameters is conditional on  
 1585 support in the WBEM protocol for filter queries in pulled instance enumeration operations.

1586 If the WBEM protocol supports filter queries in pulled instance enumeration operations, the following rules  
 1587 apply:

- 1588 • If *FilterQueryLanguage* is not NULL, additional filtering is requested and the following rules  
 1589 apply:
  - 1590 – *FilterQueryLanguage* shall specify a valid query language and *FilterQueryString* shall  
 1591 be a valid query in that query language. Neither the query language nor the format of  
 1592 the filter query is defined by this specification. Conformant WBEM protocols shall  
 1593 define a mechanism whereby WBEM servers can declare the set of query languages  
 1594 that are valid for *FilterQueryLanguage*.

- 1595 – A filter query may specify any result set (e.g., SELECT list), but because the purpose  
 1596 of the filter query is to restrict the set of instances about which information is returned,  
 1597 its result set shall be ignored. The filter query shall not define any ordering criteria.  
 1598 The filter query shall not define any grouping of objects. Operations using filter queries  
 1599 may specify additional constraints on the filter query.
- 1600 – If the WBEM server infrastructure does not support filtered enumerations, the WBEM  
 1601 server shall return failure with message WIPG0237 (Filter queries not supported by  
 1602 WBEM server infrastructure).
- 1603 – If the CIM class implementation does not support filtered enumerations, the WBEM  
 1604 server shall return failure with message WIPG0244 (Filter queries not supported by  
 1605 class implementation).
- 1606 • If *FilterQueryLanguage* is NULL, no additional filtering shall take place, and *FilterQueryString*  
 1607 shall be NULL.
  - 1608 – If *FilterQueryString* is not NULL, the WBEM server shall return failure with message  
 1609 WIPG0208 (Invalid operation input parameter value).
- 1610 If the WBEM protocol does not support filter queries in pulled instance enumeration operations, no  
 1611 additional filtering shall take place.

#### 1612 6.5.2.4 OperationTimeout

1613 The *OperationTimeout* operation input parameter determines the "operation timeout". The operation  
 1614 timeout is the minimum time the WBEM server shall maintain the open enumeration session after the last  
 1615 Open or Pull operation (unless the enumeration session was closed during that last operation). If the  
 1616 operation timeout is exceeded, the WBEM server may close the enumeration session at any time,  
 1617 releasing any possibly allocated resources related to the enumeration session.

1618 Support for the *OperationTimeout* operation parameter in a conformant WBEM protocol is mandatory.

1619 An *OperationTimeout* of 0 means that there is no operation timeout, i.e., the enumeration session is never  
 1620 closed based on time.

1621 If *OperationTimeout* is NULL, the WBEM server shall choose an operation timeout.

1622 All other values for *OperationTimeout* specify the operation timeout in seconds.

1623 A WBEM server may restrict the set of allowable values for *OperationTimeout*. This specifically includes  
 1624 the possibility for the WBEM server to not allow 0 (no timeout). If the specified value is not an allowable  
 1625 value, the WBEM server shall return failure with error message WIPG0242 (Invalid timeout). Conformant  
 1626 WBEM protocols shall define a mechanism whereby WBEM servers can declare the allowable values for  
 1627 *OperationTimeout*.

#### 1628 6.5.2.5 ContinueOnError

1629 The *ContinueOnError* operation input parameter, if TRUE, requests continuation on error. Continuation on  
 1630 error is the ability to resume an enumeration session successfully after a Pull operation that returned an  
 1631 error. A conformant WBEM server may support continuation on error. Conformant WBEM protocols shall  
 1632 define a mechanism whereby WBEM servers can declare support for continuation on error.

1633 Support for the *ContinueOnError* operation parameter is conditional on support in the WBEM protocol for  
 1634 client side control of continuation on error for pulled instance enumeration operations.

1635 If the WBEM protocol supports client side control of continuation on error for pulled instance enumeration  
 1636 operations, the following rules apply:

- 1637       • If a WBEM server does not support continuation on error and if *ContinueOnError* is TRUE, it  
1638 shall return failure with error message WIPG0235 (Continuation on error not supported).
- 1639       • If a WBEM server supports continuation on error, it shall support it as follows: If  
1640 *ContinueOnError* is TRUE, the enumeration session shall remain open when a Pull operation  
1641 returns failure, and any subsequent successful Pull operations shall return the set of elements  
1642 that would have been returned if the failing Pull operations had been successful, subject to the  
1643 consistency rules defined in 5.8. If *ContinueOnError* is FALSE, the enumeration session shall  
1644 be closed when a Pull operation returns failure.

1645 If the WBEM protocol does not support client side control of continuation on error for pulled instance  
1646 enumeration operations, it shall define requirements for the behavior of the WBEM server with respect to  
1647 continuation on error.

#### 1648 **6.5.2.6 MaxObjectCount**

1649 NOTE: This operation output parameter is also used for Pull operations.

1650 The *MaxObjectCount* operation input parameter defines the maximum number of objects that may be  
1651 returned by this operation. Any uint32 number is valid, including 0. The WBEM server may deliver any  
1652 number of objects up to *MaxObjectCount* but shall not deliver more than *MaxObjectCount* objects.

1653 Support for the *MaxObjectCount* operation parameter in a conformant WBEM protocol is mandatory.

1654 A conformant WBEM server implementation may choose to never return any elements during an  
1655 operation, regardless of the value of *MaxObjectCount*.

1656 A WBEM client may use a *MaxObjectCount* value of 0 to specify that it does not want to retrieve any  
1657 instances in the operation.

### 1658 **6.5.3 OpenClassInstancesWithPath**

#### 1659 **Purpose:**

1660 Establish and open an enumeration session for enumerating the instances of a class (including  
1661 instances of its subclasses), and optionally retrieve a first set of instances.

#### 1662 **Operation Input Parameters:**

1663

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

1664 **Operation Output Parameters:**  
1665

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

1666 **Description:**

1667 The *OpenClassInstancesWithPath* operation establishes and opens an enumeration session for  
1668 enumerating all CIM instances of the class referenced by *EnumClassPath*, including instances of any  
1669 of its subclasses. Retrieval of a first set of those instances together with their instance paths may be  
1670 requested by setting *MaxObjectCount* to a value > 0.

1671 The set of instances to be returned throughout the entire enumeration session shall be determined  
1672 using the following algorithm:

- 1673 • Initially, the set of instances to be returned is the set of instances in the namespace  
1674 specified in *EnumClassPath*, whose creation class is the class specified in *EnumClassPath*  
1675 or a subclass of that class.
- 1676 • If the WBEM protocol supports filter queries for pulled instance enumeration operations  
1677 (that is, the *FilterQueryString* and *FilterQueryLanguage* operation parameters) and  
1678 *FilterQueryLanguage* is not NULL, *FilterQueryString* acts as a restricting filter on the

1679 instances to be returned such that any instances not selected by the filter query for its  
 1680 result set are removed from the set of instances. The filter query shall query only the class  
 1681 specified in *EnumClassPath*. See also 6.5.2.3.

1682 The set of instances to be returned throughout the entire enumeration session should not contain  
 1683 any duplicate instances, as defined in 5.8.4. Because the set of returned instances contains only  
 1684 instances that exist in the same namespace, a determination of duplicate instances (for example by  
 1685 a WBEM client) can be done on the basis of their model paths only.

1686 The set of instances to be returned in the *InstanceList* operation parameter is the first set of  
 1687 instances from the set of instances to be returned throughout the entire enumeration session, such  
 1688 that no more than *MaxObjectCount* instances are returned. Returning no instances does not imply  
 1689 that the enumeration session has been exhausted. Only the *EndOfSequence* operation output  
 1690 parameter indicates whether the enumeration session has been exhausted.

1691 The set of properties to be included in any returned instances shall be determined using the following  
 1692 algorithm:

- 1693 • Initially, the set of properties to be included is the set of properties exposed by the creation  
 1694 class of the instance. This includes all the duplicates of any duplicate non-overridden  
 1695 properties.
- 1696 • If the *IncludedProperties* operation input parameter is supported by the WBEM protocol  
 1697 and if its value is not NULL, it acts as a restricting filter on the properties to be included in  
 1698 the returned instances such that any properties exposed by the creation class of the  
 1699 instance that are not named in that operation parameter are removed from the set of  
 1700 properties to be included. Any duplicate or invalid property names in the  
 1701 *IncludedProperties* operation input parameter shall be ignored. A non-NULL empty  
 1702 *IncludedProperties* list removes all properties from the set of properties to be included.
- 1703 • If the *ExcludeSubclassProperties* operation input parameter is supported by the WBEM  
 1704 protocol and if its value is TRUE, it acts as a restricting filter on the properties to be  
 1705 included in the returned instances such that any properties not exposed by the class  
 1706 referenced by *EnumClassPath* are removed from the set of properties to be included. In  
 1707 other words, the set of properties is restricted to the properties exposed by the  
 1708 enumeration class.
- 1709 • Conformant WBEM protocols may specify rules that cause properties with a value of NULL  
 1710 to be removed from the set of properties to be included.

1711 **Preconditions:**

- 1712 • The CIM class referenced by *EnumClassPath* shall exist in the namespace. If this is not  
 1713 satisfied, the operation shall fail, indicating WIPG0214.
- 1714 • If a filter query is specified,
  - 1715 – the query language specified in the *FilterQueryLanguage* operation parameter shall be  
 1716 valid. If this is not satisfied, the operation shall fail, indicating WIPG0221.
  - 1717 – the query specified in the *FilterQueryString* operation parameter shall be a valid query in  
 1718 the query language specified in the *FilterQueryLanguage* operation parameter. If this is not  
 1719 satisfied, the operation shall fail, indicating WIPG0222 or WIPG0223.

1720 **Postconditions:**

- 1721 • The enumeration session shall have been established and opened.
- 1722 • A first set of instances with their instance paths shall have been returned as described in the  
 1723 Description paragraph for this operation.

- 1724 • Requirements on ACID properties:
  - 1725 – Atomicity: Required (related to the creation of an enumeration context that is maintained by the WBEM server)
  - 1726
  - 1727 – Update Consistency: N/A
  - 1728 – Isolation: Required at the level of single instances, as defined in 5.8.
  - 1729 – Durability: Required (related to creation of an enumeration context that is maintained by the WBEM server)
  - 1730

1731 **Error Messages:**

1732

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.	

#### 1733 6.5.4 OpenClassInstancePaths

##### 1734 Purpose:

1735 Establish and open an enumeration session for enumerating the instances of a class (including  
1736 instances of its subclasses), and optionally retrieve a first set of instance paths of those instances.

##### 1737 Operation Input Parameters:

1738

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

##### 1739 Operation Output Parameters:

1740

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

1741 **Description:**

1742 The *OpenClassInstancePaths* operation establishes and opens an enumeration session for  
 1743 enumerating the CIM instance paths of all instances of the class referenced by *EnumClassPath*,  
 1744 including of instances of any of its subclasses. Retrieval of a first set of those instance paths may be  
 1745 requested by setting *MaxObjectCount* to a value > 0.

1746 The set of instances from which instance paths are to be returned throughout the entire enumeration  
 1747 session shall be determined using the following algorithm:

- 1748 • Initially, the set of instances to be returned is the set of instances in the namespace  
 1749 specified in *EnumClassPath*, whose creation class is the class specified in *EnumClassPath*  
 1750 or a subclass of that class.
- 1751 • If the WBEM protocol supports filter queries for pulled instance enumeration operations  
 1752 (that is, the *FilterQueryString* and *FilterQueryLanguage* operation parameters) and  
 1753 *FilterQueryLanguage* is not NULL, *FilterQueryString* acts as a restricting filter on the  
 1754 instances to be returned such that any instances not selected by the filter query for its  
 1755 result set are removed from the set of instances. The filter query shall query only the class  
 1756 specified in *EnumClassPath*. See also 6.5.2.3.

1757 The set of instance paths to be returned throughout the entire enumeration session should not  
 1758 contain any duplicate instance paths, as defined in 5.8.4. Because the instances referenced by the  
 1759 set of returned instance paths contains only instances that exist in the same namespace, a  
 1760 determination of duplicate instance paths can be done on the basis of their model paths only.

1761 The set of instance paths to be returned in the *InstancePathList* operation parameter is the first set of  
 1762 instance paths from the set of instance paths to be returned throughout the entire enumeration  
 1763 session, such that no more than *MaxObjectCount* instance paths are returned. Returning no instance  
 1764 paths does not imply that the enumeration session has been exhausted. Only the *EndOfSequence*  
 1765 operation output parameter indicates whether the enumeration session has been exhausted.

1766 **Preconditions:**

- 1767 • The CIM class referenced by *EnumClassPath* shall exist in the namespace. If this is not  
 1768 satisfied, the operation shall fail, indicating WIPG0214.
- 1769 • If a filter query is specified,
  - 1770 – the query language specified in the *FilterQueryLanguage* operation parameter shall be  
 1771 valid. If this is not satisfied, the operation shall fail, indicating WIPG0221.
  - 1772 – the query specified in the *FilterQueryString* operation parameter shall be a valid query in  
 1773 the query language specified in the *FilterQueryLanguage* operation parameter. If this is not  
 1774 satisfied, the operation shall fail, indicating WIPG0222 or WIPG0223.

1775 **Postconditions:**

- 1776 • The enumeration session shall have been established and opened.
- 1777 • A first set of instance paths shall have been returned as described in the Description paragraph  
 1778 for this operation.

- 1779       • Requirements on ACID properties:
- 1780       – Atomicity: Required (related to the creation of an enumeration context that is maintained by  
1781       the WBEM server)
- 1782       – Update Consistency: N/A
- 1783       – Isolation: Required at the level of single instances, as defined in 5.8.
- 1784       – Durability: Required (related to creation of an enumeration context that is maintained by  
1785       the WBEM server)

1786       **Error Messages:**

1787

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.	

1788 **6.5.5 OpenAssociatedInstancesWithPath**

1789 **Purpose:**

1790 Establish and open an enumeration session for enumerating instances that are associated with a  
 1791 given source instance, and optionally retrieve a first set of those instances.

1792 **Operation Input Parameters:**

1793

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

1794  
1795**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

1796

**Description:**

1797 The *OpenAssociatedInstancesWithPath* operation establishes and opens an enumeration session  
 1798 for enumerating instances that are associated with the specified source instance. Retrieval of a first  
 1799 set of those instances together with their instance paths may be requested by setting  
 1800 *MaxObjectCount* to a value > 0.

1801 The set of instances to be returned throughout the entire enumeration session shall be determined  
 1802 using the following algorithm:

1803 • Initially, the set of instances to be returned is the set of all instances associated to the  
 1804 source instance specified in *SourceInstancePath*. These associations may be instances of  
 1805 different association classes.

1806 The result set should not contain any duplicate instances, as defined in 5.8.4. However,  
 1807 different far ends may reference the same instance, and in such cases, the instance shall  
 1808 be contained in the result set once for each such reference.

1809 • If the *AssociationClassName* operation input parameter is not NULL, it acts as a restricting  
 1810 filter on the instances to be returned such that each instance that is associated with the  
 1811 source instance using an association whose creation class or one of its superclasses does  
 1812 not have the name specified in *AssociationClassName*, is removed from the set of  
 1813 instances to be returned. There shall be no validity checking performed for the  
 1814 *AssociationClassName* operation input parameter.

- 1815 • If the *AssociatedClassName* operation input parameter is not NULL, it acts as a restricting  
1816 filter on the instances to be returned such that each instance whose creation class or one  
1817 of its superclasses does not have the name specified in *AssociatedClassName*, is removed  
1818 from the set of instances to be returned. There shall be no validity checking performed for  
1819 the *AssociatedClassName* operation input parameter.
- 1820 • If the *SourceRoleName* operation input parameter is not NULL, it acts as a restricting filter  
1821 on the instances to be returned such that each instance that is associated with the source  
1822 instance using an association class that has a role name on the source end that is not the  
1823 role name specified in *SourceRoleName*, is removed from the set of instances to be  
1824 returned. There shall be no validity checking performed for the *SourceRoleName* operation  
1825 input parameter.
- 1826 • If the *AssociatedRoleName* operation input parameter is not NULL, it acts as a restricting  
1827 filter on the instances to be returned such that each instance that is associated with the  
1828 source instance using an association class that has a role name on the end referencing  
1829 that instance that is not the role name specified in *AssociatedRoleName*, is removed from  
1830 the set of instances to be returned. There shall be no validity checking performed for the  
1831 *AssociatedRoleName* operation input parameter.
- 1832 • If the WBEM protocol supports filter queries for pulled instance enumeration operations  
1833 (that is, the *FilterQueryString* and *FilterQueryLanguage* operation parameters) and  
1834 *FilterQueryLanguage* is not NULL, *FilterQueryString* acts as a restricting filter on the  
1835 instances to be returned such that any instances not selected by the filter query for its  
1836 result set are removed from the set of instances. The filter query shall query only the class  
1837 specified in *AssociatedClassName* (e.g., in the CQL FROM-clause). See also 6.5.2.3.
- 1838 The set of instances to be returned throughout the entire enumeration session should not contain  
1839 any duplicate instances, as defined in 5.8.4. Because the set of returned instances contains only  
1840 instances that exist in the same namespace, a determination of duplicate instances can be done on  
1841 the basis of their model paths only.
- 1842 The set of instances to be returned in the *InstanceList* operation parameter is the first set of  
1843 instances from the set of instances to be returned throughout the entire enumeration session, such  
1844 that no more than *MaxObjectCount* instances are returned. Returning no instances does not imply  
1845 that the enumeration session has been exhausted. Only the *EndOfSequence* operation output  
1846 parameter indicates whether the enumeration session has been exhausted.
- 1847 The set of properties to be included in any returned instances shall be determined using the following  
1848 algorithm:
- 1849 • Initially, the set of properties to be included is the set of properties exposed by the creation  
1850 class of the instance. This includes all the duplicates of any duplicate non-overridden  
1851 properties.
- 1852 • If the *IncludedProperties* operation input parameter is supported by the WBEM protocol  
1853 and if its value is not NULL, it acts as a restricting filter on the properties to be included in  
1854 the returned instances such that any properties exposed by the creation class of the  
1855 instance that are not named in that operation parameter are removed from the set of  
1856 properties to be included. Any duplicate or invalid property names in the  
1857 *IncludedProperties* operation input parameter shall be ignored. A non-NULL empty  
1858 *IncludedProperties* list removes all properties from the set of properties to be included.
- 1859 • If the *ExcludeSubclassProperties* operation input parameter is supported by the WBEM  
1860 protocol and if its value is TRUE, it acts as a restricting filter on the properties to be  
1861 included in the returned instances such that any properties not exposed by the class  
1862 specified in *AssociatedClassName* are removed from the set of properties to be included.

- 1863                   • Conformant WBEM protocols may specify rules that cause properties with a value of NULL  
1864                   to be removed from the set of properties to be included.

1865 **Preconditions:**

- 1866                   • The instance referenced by *SourceInstancePath* shall exist in the namespace. If this is not  
1867                   satisfied, the operation shall fail, indicating WIPG0213.
- 1868                   • If a filter query is specified,
- 1869                   – the query language specified in the *FilterQueryLanguage* operation parameter shall be  
1870                   valid. If this is not satisfied, the operation shall fail, indicating WIPG0221.
- 1871                   – the query specified in the *FilterQueryString* operation parameter shall be a valid query in  
1872                   the query language specified in the *FilterQueryLanguage* operation parameter. If this is not  
1873                   satisfied, the operation shall fail, indicating WIPG0222 or WIPG0223.
- 1874                   – the *AssociatedClassName* operation input parameter shall be non-NULL. If this is not  
1875                   satisfied, the operation shall fail, indicating WIPG0208.
- 1876                   • The *IncludedProperties* operation parameter, if supported by the WBEM protocol, shall only be  
1877                   specified with a non-NULL value if the *AssociatedClassName* operation input parameter is also  
1878                   non-NULL. If this is not satisfied, the operation shall fail, indicating WIPG0208.
- 1879                   • The *ExcludeSubclassProperties* operation parameter, if supported by the WBEM protocol, shall  
1880                   only be specified with a TRUE value if the *AssociatedClassName* operation input parameter is  
1881                   non-NULL. If this is not satisfied, the operation shall fail, indicating WIPG0208.

1882                   NOTE: Specifying a non-NULL value for *AssociatedClassName* ensures that the associated instances have the  
1883                   class specified in *AssociatedClassName* as a common superclass.

1884 **Postconditions:**

- 1885                   • The enumeration session shall have been established and opened.
- 1886                   • A first set of instances with their instance paths shall have been returned as described in the  
1887                   Description paragraph for this operation.
- 1888                   • Requirements on ACID properties:
- 1889                   – Atomicity: Required (related to the creation of an enumeration context that is maintained by  
1890                   the WBEM server)
- 1891                   – Update Consistency: N/A
- 1892                   – Isolation: Required at the level of single instances, as defined in 5.8.
- 1893                   – Durability: Required (related to creation of an enumeration context that is maintained by  
1894                   the WBEM server)

1895 **Error Messages:**

1896

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.	

1897 **6.5.6 OpenAssociatedInstancePaths**

1898 **Purpose:**

1899 Establish and open an enumeration session for enumerating the instance paths of instances that are  
 1900 associated with a given source instance, and optionally retrieve a first set of those instance paths.

1901 **Operation Input Parameters:**

1902

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

1903  
1904**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

1905 **Description:**

1906 The *OpenAssociatedInstancePaths* operation establishes and opens an enumeration session for  
 1907 enumerating the instance paths of instances that are associated with the specified source instance.  
 1908 Retrieval of a first set of those instance paths may be requested by setting *MaxObjectCount* to a  
 1909 value > 0.

1910 The set of instances of which instance paths are to be returned throughout the entire enumeration  
 1911 session shall be determined using the following algorithm:

- 1912 • Initially, the set of instances to be returned is the set of all instances associated to the  
 1913 source instance specified in *SourceInstancePath*. These associations may be instances of  
 1914 different association classes.

1915 The result set should not contain any duplicate instance paths, as defined in 5.8.4.  
 1916 However, different far ends may reference the same instance, and in such cases, the  
 1917 instance path shall be contained in the result set once for each such reference.

- 1918 • If the *AssociationClassName* operation input parameter is not NULL, it acts as a restricting  
 1919 filter on the instances to be returned such that each instance that is associated with the  
 1920 source instance using an association whose creation class or one of its superclasses does  
 1921 not have the name specified in *AssociationClassName*, is removed from the set of  
 1922 instances to be returned. There shall be no validity checking performed for the  
 1923 *AssociationClassName* operation input parameter.

- 1924 • If the *AssociatedClassName* operation input parameter is not NULL, it acts as a restricting  
 1925 filter on the instances to be returned such that each instance whose creation class or one  
 1926 of its superclasses does not have the name specified in *AssociatedClassName*, is removed  
 1927 from the set of instances to be returned. There shall be no validity checking performed for  
 1928 the *AssociatedClassName* operation input parameter.

- 1929 • If the *SourceRoleName* operation input parameter is not NULL, it acts as a restricting filter  
 1930 on the instances to be returned such that each instance that is associated with the source  
 1931 instance using an association class that has a role name on the source end that is not the  
 1932 role name specified in *SourceRoleName*, is removed from the set of instances to be  
 1933 returned. There shall be no validity checking performed for the *SourceRoleName* operation  
 1934 input parameter.

- 1935 • If the *AssociatedRoleName* operation input parameter is not NULL, it acts as a restricting  
 1936 filter on the instances to be returned such that each instance that is associated with the  
 1937 source instance using an association class that has a role name on the end referencing  
 1938 that instance that is not the role name specified in *AssociatedRoleName*, is removed from  
 1939 the set of instances to be returned. There shall be no validity checking performed for the  
 1940 *AssociatedRoleName* operation input parameter.

- 1941 • If the WBEM protocol supports filter queries for pulled instance enumeration operations  
 1942 (that is, the *FilterQueryString* and *FilterQueryLanguage* operation parameters) and  
 1943 *FilterQueryLanguage* is not NULL, *FilterQueryString* acts as a restricting filter on the  
 1944 instances to be returned such that any instances not selected by the filter query for its  
 1945 result set are removed from the set of instances. The filter query shall query only the class  
 1946 specified in *AssociatedClassName* (e.g., in the CQL FROM-clause). See also 6.5.2.3.

1947 The set of instance paths to be returned throughout the entire enumeration session should not  
 1948 contain any duplicate instance paths, as defined in 5.8.4. Because the set of returned instance paths  
 1949 references only instances in the same namespace, a determination of duplicate instance paths can  
 1950 be done on the basis of their model paths only.

1951 The set of instance paths to be returned in the *InstancePathList* operation parameter is the first set of  
 1952 instance paths from the set of instance paths to be returned throughout the entire enumeration

1953 session, such that no more than *MaxObjectCount* instance paths are returned. Returning no instance  
 1954 paths does not imply that the enumeration session has been exhausted. Only the *EndOfSequence*  
 1955 operation output parameter indicates whether the enumeration session has been exhausted.

1956 **Preconditions:**

- 1957 • The instance referenced by *SourceInstancePath* shall exist in the namespace. If this is not  
 1958 satisfied, the operation shall fail, indicating WIPG0213.
- 1959 • If a filter query is specified,
  - 1960 – the query language specified in the *FilterQueryLanguage* operation parameter shall be  
 1961 valid. If this is not satisfied, the operation shall fail, indicating WIPG0221.
  - 1962 – the query specified in the *FilterQueryString* operation parameter shall be a valid query in  
 1963 the query language specified in the *FilterQueryLanguage* operation parameter. If this is not  
 1964 satisfied, the operation shall fail, indicating WIPG0222 or WIPG0223.
  - 1965 – the *AssociatedClassName* operation input parameter shall be non-NULL. If this is not  
 1966 satisfied, the operation shall fail, indicating WIPG0208.

1967 NOTE: Specifying a non-NULL value for *AssociatedClassName* ensures that the associated instances have the  
 1968 class specified in *AssociatedClassName* as a common superclass.

1969 **Postconditions:**

- 1970 • The enumeration session shall have been established and opened.
- 1971 • A first set of instance paths shall have been returned as described in the Description paragraph  
 1972 for this operation.
- 1973 • Requirements on ACID properties:
  - 1974 – Atomicity: Required (related to the creation of an enumeration context that is maintained by  
 1975 the WBEM server)
  - 1976 – Update Consistency: N/A
  - 1977 – Isolation: Required at the level of single instances, as defined in 5.8.
  - 1978 – Durability: Required (related to creation of an enumeration context that is maintained by  
 1979 the WBEM server)

1980 **Error Messages:**

1981

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.	

1982 **6.5.7 OpenReferencingInstancesWithPath**

1983 **Purpose:**

1984 Establish and open an enumeration session for enumerating the association instances that reference  
 1985 a given source instance, and optionally retrieve a first set of those instances.

1986 **Operation Input Parameters:**

1987

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

1988  
1989**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

1990 **Description:**

1991 The *OpenReferencingInstancesWithPath* operation establishes and opens an enumeration session  
 1992 for enumerating the association instances that reference the specified source instance. Retrieval of a  
 1993 first set of those instances together with their instance paths may be requested by setting  
 1994 *MaxObjectCount* to a value > 0.

1995 The set of instances to be returned throughout the entire enumeration session shall be determined  
 1996 using the following algorithm:

- 1997 • Initially, the set of instances to be returned is the set of all instances referencing the source  
 1998 instance specified in *SourceInstancePath*. These associations may be instances of  
 1999 different association classes.
- 2000 • If the *AssociationClassName* operation input parameter is not NULL, it acts as a restricting  
 2001 filter on the instances to be returned such that each association instance whose creation  
 2002 class or one of its superclasses does not have the name specified in  
 2003 *AssociationClassName*, is removed from the set of instances to be returned. There shall be  
 2004 no validity checking performed for the *AssociationClassName* operation input parameter.
- 2005 • If the *AssociatedClassName* operation input parameter is not NULL, it acts as a restricting  
 2006 filter on the instances to be returned such that each association instance whose creation  
 2007 class has a set of far ends none of which is referencing a class where that class or one of  
 2008 its superclasses has the name specified in *AssociatedClassName*, is removed from the set  
 2009 of instances to be returned. There shall be no validity checking performed for the  
 2010 *AssociatedClassName* operation input parameter.
- 2011 • If the *SourceRoleName* operation input parameter is not NULL, it acts as a restricting filter  
 2012 on the instances to be returned such that each association instance whose creation class  
 2013 does not have the role name specified in *SourceRoleName* on the end referencing the  
 2014 source instance, is removed from the set of instances to be returned. There shall be no  
 2015 validity checking performed for the *SourceRoleName* operation input parameter.
- 2016 • If the *AssociatedRoleName* operation input parameter is not NULL, it acts as a restricting  
 2017 filter on the instances to be returned such that each association instance whose creation  
 2018 class has a set of far ends none of which has the role name specified in  
 2019 *AssociatedRoleName*, is removed from the set of instances to be returned. There shall be  
 2020 no validity checking performed for the *AssociatedRoleName* operation input parameter.
- 2021 • If the WBEM protocol supports filter queries for pulled instance enumeration operations  
 2022 (that is, the *FilterQueryString* and *FilterQueryLanguage* operation parameters) and  
 2023 *FilterQueryLanguage* is not NULL, *FilterQueryString* acts as a restricting filter on the  
 2024 instances to be returned such that any instances not selected by the filter query for its  
 2025 result set are removed from the set of instances. The filter query shall query only the class  
 2026 specified in *AssociationClassName* (e.g., in the CQL FROM-clause). See also 6.5.2.3.

2027 The set of instances to be returned throughout the entire enumeration session should not contain  
 2028 any duplicate instances, as defined in 5.8.4. Because the set of returned instances contains only  
 2029 instances that exist in the same namespace, so any determination of duplicate instances (for  
 2030 example by a WBEM client) may be done on the basis of their model paths.

2031 The set of instances to be returned in the *InstanceList* operation parameter is the first set of  
 2032 instances from the set of instances to be returned throughout the entire enumeration session, such

2033 that no more than *MaxObjectCount* instances are returned. Returning no instances does not imply  
 2034 that the enumeration session has been exhausted. Only the *EndOfSequence* operation output  
 2035 parameter indicates whether the enumeration session has been exhausted.

2036 The set of properties to be included in any returned instances shall be determined using the following  
 2037 algorithm:

- 2038 • Initially, the set of properties to be included is the set of properties exposed by the creation  
 2039 class of the instance. This includes all the duplicates of any duplicate non-overridden  
 2040 properties.
- 2041 • If the *IncludedProperties* operation input parameter is supported by the WBEM protocol  
 2042 and if its value is not NULL, it acts as a restricting filter on the properties to be included in  
 2043 the returned instances such that any properties exposed by the creation class of the  
 2044 instance that are not named in that operation parameter are removed from the set of  
 2045 properties to be included. Any duplicate or invalid property names in the  
 2046 *IncludedProperties* operation input parameter shall be ignored. A non-NULL empty  
 2047 *IncludedProperties* list removes all properties from the set of properties to be included.
- 2048 • If the *ExcludeSubclassProperties* operation input parameter is supported by the WBEM  
 2049 protocol and if its value is TRUE, it acts as a restricting filter on the properties to be  
 2050 included in the returned instances such that any properties not exposed by the class  
 2051 specified in *AssociationClassName* are removed from the set of properties to be included.
- 2052 • Conformant WBEM protocols may specify rules that cause properties with a value of NULL  
 2053 to be removed from the set of properties to be included.

#### 2054 **Preconditions:**

- 2055 • The instance referenced by *SourceInstancePath* shall exist in the namespace. If this is not  
 2056 satisfied, the operation shall fail, indicating WIPG0213.
- 2057 • If a filter query is specified,
  - 2058 – the query language specified in the *FilterQueryLanguage* operation parameter shall be  
 2059 valid. If this is not satisfied, the operation shall fail, indicating WIPG0221.
  - 2060 – the query specified in the *FilterQueryString* operation parameter shall be a valid query in  
 2061 the query language specified in the *FilterQueryLanguage* operation parameter. If this is not  
 2062 satisfied, the operation shall fail, indicating WIPG0222 or WIPG0223.
  - 2063 – the *AssociationClassName* operation input parameter shall be non-NULL. If this is not  
 2064 satisfied, the operation shall fail, indicating WIPG0208.
- 2065 • The *IncludedProperties* operation parameter, if supported by the WBEM protocol, shall only be  
 2066 specified with a non-NULL value if the *AssociationClassName* operation input parameter is also  
 2067 non-NULL. If this is not satisfied, the operation shall fail, indicating WIPG0208.
- 2068 • The *ExcludeSubclassProperties* operation parameter, if supported by the WBEM protocol, shall  
 2069 only be specified with a TRUE value if the *AssociationClassName* operation input parameter is  
 2070 non-NULL. If this is not satisfied, the operation shall fail, indicating WIPG0208.

2071 NOTE: Specifying a non-NULL value for *AssociationClassName* ensures that the association instances have the  
 2072 class specified in *AssociationClassName* as a common superclass.

#### 2073 **Postconditions:**

- 2074 • The enumeration session shall have been established and opened.
- 2075 • A first set of instances with their instance paths shall have been returned as described in the  
 2076 Description paragraph for this operation.
- 2077 • Requirements on ACID properties:

- 2078           – Atomicity: Required (related to the creation of an enumeration context that is maintained by the WBEM server)
- 2079
- 2080           – Update Consistency: N/A
- 2081           – Isolation: Required at the level of single instances, as defined in 5.8.
- 2082           – Durability: Required (related to creation of an enumeration context that is maintained by the WBEM server)
- 2083

2084 **Error Messages:**  
2085

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.	

2086 **6.5.8 OpenReferencingInstancePaths**2087 **Purpose:**

2088 Establish and open an enumeration session for enumerating the instance paths of association  
 2089 instances that reference a given source instance, and optionally retrieve a first set of those instance  
 2090 paths.

2091 **Operation Input Parameters:**  
2092

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

2093  
2094

**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

2095

**Description:**

2096 The *OpenReferencingInstancePaths* operation establishes and opens an enumeration session for  
 2097 enumerating the instance paths of association instances that reference the specified source  
 2098 instance. Retrieval of a first set of those instance paths may be requested by setting  
 2099 *MaxObjectCount* to a value > 0.

2100 The set of instances of which instance paths are to be returned throughout the entire enumeration  
 2101 session shall be determined using the following algorithm:

- 2102 • Initially, the set of instances to be returned is the set of all instances referencing the source  
 2103 instance specified in *SourceInstancePath*. These associations may be instances of  
 2104 different association classes.
- 2105 • If the *AssociationClassName* operation input parameter is not NULL, it acts as a restricting  
 2106 filter on the instances to be returned such that each association instance whose creation  
 2107 class or one of its superclasses does not have the name specified in  
 2108 *AssociationClassName*, is removed from the set of instances to be returned. There shall be  
 2109 no validity checking performed for the *AssociationClassName* operation input parameter.
- 2110 • If the *AssociatedClassName* operation input parameter is not NULL, it acts as a restricting  
 2111 filter on the instances to be returned such that each association instance whose creation  
 2112 class has a set of far ends none of which is referencing a class where that class or one of  
 2113 its superclasses has the name specified in *AssociatedClassName*, is removed from the set  
 2114 of instances to be returned. There shall be no validity checking performed for the  
 2115 *AssociatedClassName* operation input parameter.
- 2116 • If the *SourceRoleName* operation input parameter is not NULL, it acts as a restricting filter  
 2117 on the instances to be returned such that each association instance whose creation class  
 2118 does not have the role name specified in *SourceRoleName* on the end referencing the  
 2119 source instance, is removed from the set of instances to be returned. There shall be no  
 2120 validity checking performed for the *SourceRoleName* operation input parameter.
- 2121 • If the *AssociatedRoleName* operation input parameter is not NULL, it acts as a restricting  
 2122 filter on the instances to be returned such that each association instance whose creation  
 2123 class has a set of far ends none of which has the role name specified in  
 2124 *AssociatedRoleName*, is removed from the set of instances to be returned. There shall be  
 2125 no validity checking performed for the *AssociatedRoleName* operation input parameter.

- 2126
- 2127
- 2128
- 2129
- 2130
- 2131
- If the WBEM protocol supports filter queries for pulled instance enumeration operations (that is, the *FilterQueryString* and *FilterQueryLanguage* operation parameters) and *FilterQueryLanguage* is not NULL, *FilterQueryString* acts as a restricting filter on the instances to be returned such that any instances not selected by the filter query for its result set are removed from the set of instances. The filter query shall query only the class specified in *AssociationClassName* (e.g., in the CQL FROM-clause). See also 6.5.2.3.

2132

2133

2134

2135

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

2136

2137

2138

2139

2140

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

2141

**Preconditions:**

- 2142
- 2143
- 2144
- 2145
- 2146
- 2147
- 2148
- 2149
- 2150
- 2151
- The instance referenced by *SourceInstancePath* shall exist in the namespace. If this is not satisfied, the operation shall fail, indicating WIPG0213.
  - If a filter query is specified,
    - the query language specified in the *FilterQueryLanguage* operation parameter shall be valid. If this is not satisfied, the operation shall fail, indicating WIPG0221.
    - the query specified in the *FilterQueryString* operation parameter shall be a valid query in the query language specified in the *FilterQueryLanguage* operation parameter. If this is not satisfied, the operation shall fail, indicating WIPG0222 or WIPG0223.
    - the *AssociationClassName* operation input parameter shall be non-NULL. If this is not satisfied, the operation shall fail, indicating WIPG0208.

2152

2153

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

2154

**Postconditions:**

- 2155
- 2156
- 2157
- 2158
- 2159
- 2160
- 2161
- 2162
- 2163
- 2164
- The enumeration session shall have been established and opened.
  - A first set of instance paths shall have been returned as described in the Description paragraph for this operation.
  - Requirements on ACID properties:
    - Atomicity: Required (related to the creation of an enumeration context that is maintained by the WBEM server)
    - Update Consistency: N/A
    - Isolation: Required at the level of single instances, as defined in 5.8.
    - Durability: Required (related to creation of an enumeration context that is maintained by the WBEM server)

2165

2166

**Error Messages:**

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	

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

## 2167 6.5.9 OpenQueryInstances

### 2168 Purpose:

2169 Establish and open an enumeration session for enumerating the instances of a query result, and  
 2170 optionally retrieve a first set of instances.

2171  
2172**Operation Input Parameters:**

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

2173  
2174**Operation Output Parameters:**

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

**2175 Description:**

2176 The *OpenQueryInstances* operation establishes and opens an enumeration session for enumerating  
2177 the instances representing the result of the query specified in *QueryString* in the CIM namespace  
2178 referenced by *NamespacePath*. Retrieval of a first set of those instances may be requested by  
2179 setting *MaxObjectCount* to a value > 0.

2180 The set of instances to be returned in the *InstanceList* operation parameter is the first set of  
2181 instances from the set of instances to be returned throughout the entire enumeration session, such  
2182 that no more than *MaxObjectCount* instances are returned. Returning no instances in the  
2183 *InstanceList* operation parameter does not imply that the enumeration session has been exhausted.  
2184 Only the *EndOfSequence* operation output parameter indicates whether the enumeration session  
2185 has been exhausted.

2186 The returned instances are only representations of instances that have no corresponding  
 2187 addressable instances residing in the WBEM server.

2188 If *QueryLanguage* is not NULL, it shall specify a valid query language and *QueryString* shall be a  
 2189 valid query in that query language. Neither the query language nor the format of the filter query is  
 2190 defined by this specification. Conformant WBEM protocols shall specify a mechanism for determining  
 2191 the set of query languages that are valid for *QueryLanguage*. The simplest way to do this is to list the  
 2192 set of valid query languages.

2193 The value of the *ReturnQueryResultClass* operation input parameter controls whether or not a class  
 2194 definition is returned in the *QueryResultClass* operation output parameter. If FALSE, then  
 2195 *QueryResultClass* shall be NULL. If TRUE, then the value of *QueryResultClass* shall be a class  
 2196 definition that defines the properties of each instance of the query result. The name of this class shall  
 2197 be CIM\_QueryResult. This class is only a representation of a class that has no corresponding  
 2198 addressable class residing in the WBEM server.

2199 **Preconditions:**

- 2200 • The CIM namespace referenced by *NamespacePath* shall exist. If this is not satisfied, the  
 2201 operation shall fail, indicating WIPG0204.
- 2202 • The query language specified in the *QueryLanguage* operation parameter shall be a valid query  
 2203 language. If this is not satisfied, the operation shall fail, indicating WIPG0221.
- 2204 • The query specified in the *QueryString* operation parameter shall be a valid query in the query  
 2205 language specified in the *QueryLanguage* operation parameter. If this is not satisfied, the  
 2206 operation shall fail, indicating WIPG0222 or WIPG0223.

2207 **Postconditions:**

- 2208 • The enumeration session shall have been established and opened.
- 2209 • A first set of instances shall have been returned as described in the Description paragraph for  
 2210 this operation.
- 2211 • Requirements on ACID properties:
  - 2212 – Atomicity: Required (related to the creation of an enumeration context that is maintained by  
 2213 the WBEM server)
  - 2214 – Update Consistency: N/A
  - 2215 – Isolation: Required at the level of single instances, as defined in 5.8.
  - 2216 – Durability: Required (related to creation of an enumeration context that is maintained by  
 2217 the WBEM server)

2218 **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	

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.	

## 2220 6.5.10 Common operation parameters for the pull operations

2221 This subclause defines commonly used operation parameters for the Pull operations. The description of  
 2222 the individual Pull operations references these operation parameters as appropriate. However, not every  
 2223 Pull operation uses every one of these common operation parameters.

### 2224 6.5.10.1 NamespacePath

2225 The *NamespacePath* operation input parameter references the CIM namespace identified by the context  
 2226 parameter of the Open operation that established and opened the enumeration session.

### 2227 6.5.10.2 EnumerationContext

2228 The *EnumerationContext* operation input/output parameter is the enumeration context value representing  
 2229 the enumeration session to be used.

2230 Support for the *EnumerationContext* operation parameter in a conformant WBEM protocol is mandatory.

2231 When invoking the Pull operation, the enumeration session represented by *EnumerationContext* shall be  
 2232 open. The enumeration session shall have been established using one of the Open operations whose  
 2233 type of enumerated element matches the Pull operation. For the first Pull operation on an enumeration  
 2234 session, the value of *EnumerationContext* shall be the enumeration context value returned by a  
 2235 successful Open operation that established and opened that enumeration session. For any subsequent  
 2236 Pull operations on that enumeration session, the value of *EnumerationContext* shall be the value of  
 2237 *EnumerationContext* as returned by the previous Pull operation on the same enumeration session.

2238 After completing the Pull operation, the enumeration session represented by *EnumerationContext* shall  
 2239 be open or closed.

2240 **6.5.10.3 EndOfSequence**

2241 The *EndOfSequence* operation output parameter when used in Pull operations behaves as defined in  
 2242 6.5.2.2

2243 **6.5.10.4 MaxObjectCount**

2244 The *MaxObjectCount* operation input parameter when used in Pull operations behaves as defined in  
 2245 6.5.2.6.

2246 **6.5.11 PullInstancesWithPath**

2247 **Purpose:**

2248 Retrieve the next set of instances together with their instance paths from an open enumeration  
 2249 session.

2250 **Operation Input Parameters:**

2251

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

2252 **Operation Output Parameters:**

2253

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

2254 **Description:**

2255 The *PullInstancesWithPath* operation retrieves the next set of instances together with their instance  
 2256 paths from an open enumeration session.

2257 The enumeration session shall have been established using one of the following operations:

- 2258 • *OpenClassInstancesWithPath*

- 2259           • OpenAssociatedInstancesWithPath
- 2260           • OpenReferencingInstancesWithPath

2261           The set of instances to be returned in the *InstanceList* operation parameter is the next set of  
 2262 instances from the set of instances to be returned throughout the entire enumeration session, such  
 2263 that no more than *MaxObjectCount* instances are returned. Returning no instances does not imply  
 2264 that the enumeration session has been exhausted. Only the *EndOfSequence* operation output  
 2265 parameter indicates whether the enumeration session has been exhausted.

2266           The set of properties to be included in any retrieved instances shall be the as determined using the  
 2267 Open operation that established the enumeration session.

2268   **Preconditions:**

- 2269           • The enumeration session identified by *EnumerationContext* shall be open. If this is not satisfied,  
 2270 the operation shall fail, indicating WIPG0241.
- 2271           • The value of *EnumerationContext* shall be the enumeration context value returned by the  
 2272 previous Open or Pull operation on the same enumeration session. If this is not satisfied, the  
 2273 operation shall fail, indicating WIPG0241.

2274   **Postconditions:**

- 2275           • The set of instances with their instance paths shall have been returned as described in the  
 2276 Description paragraph for this operation.
- 2277           • Requirements on ACID properties:
- 2278           – Atomicity: Required (related to updates to an enumeration context that is maintained by the  
 2279 WBEM server)
- 2280           – Update Consistency: N/A
- 2281           – Isolation: Required at the level of single instances, as defined in 5.8.
- 2282           – Durability: Required (related to updates to an enumeration context that is maintained by  
 2283 the WBEM server)

2284   **Error Messages:**

2285

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.	

2286 **6.5.12 PullInstancePaths**

2287 **Purpose:**

2288 Retrieve the next set of instance paths from an open enumeration session.

2289 **Operation Input Parameters:**

2290

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

2291 **Operation Output Parameters:**

2292

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

2293 **Description:**

2294 The *PullInstancePaths* operation retrieves the next set of instance paths from an open enumeration session.

2296 The enumeration session shall have been established using one of the following operations:

- 2297           •   OpenClassInstancePaths
- 2298           •   OpenAssociatedInstancePaths
- 2299           •   OpenReferencingInstancePaths

2300           The set of instance paths to be returned in the *InstancePathList* operation parameter is the next set  
 2301           of instance paths from the set of instance paths to be returned throughout the entire enumeration  
 2302           session, such that no more than *MaxObjectCount* instance paths are returned. Returning no instance  
 2303           paths does not imply that the enumeration session has been exhausted. Only the *EndOfSequence*  
 2304           operation output parameter indicates whether the enumeration session has been exhausted.

#### 2305   **Preconditions:**

- 2306           •   The enumeration session identified by *EnumerationContext* shall be open. If this is not satisfied,  
 2307           the operation shall fail, indicating WIPG0241.
- 2308           •   The value of *EnumerationContext* shall be the enumeration context value returned by the  
 2309           previous Open or Pull operation on the same enumeration session. If this is not satisfied, the  
 2310           operation shall fail, indicating WIPG0241.

#### 2311   **Postconditions:**

- 2312           •   The set of instance paths shall have been returned as described in the Description paragraph  
 2313           for this operation.
- 2314           •   Requirements on ACID properties:
  - 2315           –   Atomicity: Required (related to updates to an enumeration context that is maintained by the  
 2316           WBEM server)
  - 2317           –   Update Consistency: N/A
  - 2318           –   Isolation: Required at the level of single instances, as defined in 5.8.
  - 2319           –   Durability: Required (related to updates to an enumeration context that is maintained by  
 2320           the WBEM server)

#### 2321   **Error Messages:**

2322

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.	

2323 **6.5.13 PullInstances**

2324 **Purpose:**

2325 Retrieve the next set of instances from an open enumeration session.

2326 **Operation Input Parameters:**

2327

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

2328 **Operation Output Parameters:**

2329

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

2330 **Description:**

2331 The *PullInstancesWithPath* operation retrieves the next set of instances together with their instance  
2332 paths from an open enumeration session.

2333 The enumeration session shall have been established using one of the following operations:

- 2334 • *OpenQueryInstances*

2335 The set of instances to be returned in the *InstanceList* operation parameter is the next set of  
 2336 instances from the set of instances to be returned throughout the entire enumeration session, such  
 2337 that no more than *MaxObjectCount* instances are returned. Returning no instances does not imply  
 2338 that the enumeration session has been exhausted. Only the *EndOfSequence* operation output  
 2339 parameter indicates whether the enumeration session has been exhausted.

2340 The set of properties to be included in any retrieved instances shall be the as determined using the  
 2341 Open operation that established the enumeration session.

#### 2342 **Preconditions:**

- 2343 • The enumeration session identified by *EnumerationContext* shall be open. If this is not satisfied,  
 2344 the operation shall fail, indicating WIPG0241.
- 2345 • The value of *EnumerationContext* shall be the enumeration context value returned by the  
 2346 previous Open or Pull operation on the same enumeration session. If this is not satisfied, the  
 2347 operation shall fail, indicating WIPG0241.

#### 2348 **Postconditions:**

- 2349 • The set of instances shall have been returned as described in the Description paragraph for this  
 2350 operation.
- 2351 • Requirements on ACID properties:
  - 2352 – Atomicity: Required (related to updates to an enumeration context that is maintained by the  
 2353 WBEM server)
  - 2354 – Update Consistency: N/A
  - 2355 – Isolation: Required at the level of single instances, as defined in 5.8.
  - 2356 – Durability: Required (related to updates to an enumeration context that is maintained by  
 2357 the WBEM server)

#### 2358 **Error Messages:**

2359

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.	

2360 **6.5.14 CloseEnumeration**

2361 **Purpose:**

2362 Close an open enumeration session.

2363 **Operation Input Parameters:**

2364

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

2365 **Operation Output Parameters:**

2366 None.

2367 **Description:**

2368 The *CloseEnumeration* operation closes the open enumeration session identified by  
2369 *EnumerationContext*.

2370 The enumeration session shall have been established using any of the Open operations.

2371 Enumeration sessions are closed implicitly when exhausted, so this operation only needs to be used  
2372 when terminating an enumeration sequence before it is exhausted.

2373 **Preconditions:**

2374 • The enumeration session identified by *EnumerationContext* shall be open. If this is not satisfied,  
2375 the operation shall fail, indicating WIPG0241.

2376 • The value of *EnumerationContext* shall be the enumeration context value returned by the  
2377 previous Open or Pull operation on the same enumeration session. If this is not satisfied, the  
2378 operation shall fail, indicating WIPG0241.

2379 **Postconditions:**

2380 • The enumeration session identified by *EnumerationContext* is closed.

2381 • Requirements on ACID properties:

- 2382           – Atomicity: Required (related to updates to or deletion of an enumeration context that is  
2383 maintained by the WBEM server)
- 2384           – Update Consistency: N/A
- 2385           – Isolation: Required
- 2386           – Durability: Required (related to updates to or deletion of an enumeration context that is  
2387 maintained by the WBEM server)

2388 **Error Messages:**  
2389

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.	

2390 **6.5.15 EnumerationCount**

2391 **Purpose:**

2392 Estimate the total number of remaining items in an open enumeration session.

2393 **Operation Input Parameters:**  
2394

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

Generic Name	Generic Type	Requirement	Description
EnumerationContext	EnumerationContext	Mandatory	Enumeration context value, as defined in 6.5.10.2

2395 **Operation Output Parameters:**

2396

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

2397 **Description:**

2398 The *EnumerationCount* operation estimates the total number of remaining items in the open  
2399 enumeration session identified by *EnumerationContext*.

2400 The enumeration session shall have been established using any of the Open operations.

2401 If not NULL, the *EnumerationCount* operation output parameter is an estimated count of the number  
2402 of items remaining to be retrieved with subsequent Pull operations. Thus, executing this operation  
2403 immediately after opening the enumeration session provides an estimate of the total number of items  
2404 that will be returned in the enumeration set.

2405 If the WBEM server cannot or will not return an estimated count, it may respond with success and  
2406 the NULL value in the *EnumerationCount* operation output parameter.

2407 This mechanism is intended to assist WBEM clients in determining the overall size of an  
2408 enumeration set and of the number of items remaining in the enumeration session. However,  
2409 because it is an estimate and not an exact number, it should not be used for determining the end of  
2410 an enumeration sequence, i.e., in place of the *EndOfSequence* operation output parameter on Open  
2411 and Pull operations.

2412 **Preconditions:**

- 2413 • The enumeration session identified by *EnumerationContext* shall be open. If this is not satisfied,  
2414 the operation shall fail, indicating WIPG0241.
- 2415 • The value of *EnumerationContext* shall be the enumeration context value returned by the  
2416 previous Open or Pull operation on the same enumeration session. If this is not satisfied, the  
2417 operation shall fail, indicating WIPG0241.

2418 **Postconditions:**

- 2419 • Requirements on ACID properties:
  - 2420 – Atomicity: N/A
  - 2421 – Update Consistency: N/A
  - 2422 – Isolation: Required
  - 2423 – Durability: N/A

2424 **Error Messages:**

2425

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	

Message ID	Message Name	Requirement	Sources	Additional Description
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.	
WIPG0241	Invalid enumeration context	Mandatory	Class implem.	
WIPG0243	Timeout	Optional	Infrastructure, class implem.	
WIPG0227	Other failure	Optional	Infrastructure, class implem.	

## 2426 6.6 Method invocation

2427 This subclause defines operations for the invocation of CIM methods.

### 2428 6.6.1 InvokeMethod

#### 2429 Purpose:

2430 Invoke a CIM method using an instance path.

#### 2431 Operation Input Parameters:

2432

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

2433 **Operation Output Parameters:**  
2434

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

2435 **Description:**

2436 Invoke a CIM method using an instance path. The method may be static or non-static.

2437 Conformant WBEM protocols shall define a mapping for the invocation of CIM methods using an  
2438 instance path, including a mapping of the operation parameters defined in the tables above. These  
2439 rules may map the method invocation to a single operation, map each method to its own separate  
2440 operation, or define any other appropriate mapping.

2441 If the implementation of the method could be invoked, the operation is considered successful,  
2442 regardless of what the semantics of any return values or output parameters is. For example, if a  
2443 method defines that a particular return value indicates an error condition, the method invocation was  
2444 still successful from a perspective of the invocation operation.

2445 **Preconditions:**

- 2446 • The instance referenced by *InstancePath* shall exist in the namespace. If this is not satisfied,  
2447 the operation shall fail, indicating WIPG0213.
- 2448 • The method to be invoked shall be exposed by the creation class of the instance referenced by  
2449 *InstancePath*. If this is not satisfied, the operation shall fail, indicating WIPG0218.

2450 **Postconditions:**

- 2451 • The CIM method shall have been invoked.
- 2452 • Requirements on ACID properties:
  - 2453 – Atomicity: Recommended
  - 2454 – Update Consistency: Recommended
  - 2455 – Isolation: Recommended
  - 2456 – Durability: Required

2457 **Error Messages:**  
2458

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	

Message ID	Message Name	Requirement	Sources	Additional Description
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.	
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.	

## 2459 6.6.2 InvokeStaticMethod

### 2460 Purpose:

2461 Invoke a static CIM method using a class path.

### 2462 Operation Input Parameters:

2463

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

### 2464 Operation Output Parameters:

2465

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

### 2466 Description:

2467 Invoke a static CIM method using a class path.

2468 Conformant WBEM protocols shall define a mapping for the invocation of CIM methods using a class  
 2469 path, including a mapping of the operation parameters defined in the tables above. These rules may  
 2470 map the method invocation to a single operation, map each method to its own separate operation, or  
 2471 define any other appropriate mapping.

2472 If the implementation of the method could be invoked, the operation is considered successful,  
 2473 regardless of what the semantics of any return values or output parameters is. For example, if a  
 2474 method defines that a particular return value indicates an error condition, the method invocation was  
 2475 still successful from a perspective of the invocation operation.

2476 **Preconditions:**

- 2477 • The instance referenced by *InstancePath* shall exist in the namespace. If this is not satisfied,  
 2478 the operation shall fail, indicating WIPG0213.
- 2479 • The method to be invoked shall be exposed by the creation class of the instance referenced by  
 2480 *InstancePath*. If this is not satisfied, the operation shall fail, indicating WIPG0218.

2481 **Postconditions:**

- 2482 • The CIM method shall have been invoked.
- 2483 • Requirements on ACID properties:
  - 2484 – Atomicity: Recommended
  - 2485 – Update Consistency: Recommended
  - 2486 – Isolation: Recommended
  - 2487 – Durability: Required

2488 **Error Messages:**

2489

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.	

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0219	Method not supported by class implementation	Mandatory	Class implem.	
WIPG0243	Timeout	Optional	Infrastructure, class implem.	
WIPG0227	Other failure	Optional	Infrastructure, class implem.	

## 2490 6.7 Class operations

2491 This subclause defines class operations (operations that target a single CIM class or create a CIM class).  
 2492 These operations include dealing with qualifier values defined on classes.

### 2493 6.7.1 GetClass

#### 2494 Purpose:

2495 Retrieve a CIM class.

#### 2496 Operation Input Parameters:

2497

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

#### 2498 Operation Output Parameters:

2499

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

#### 2500 Description:

2501 The *GetClass* operation retrieves a representation of the CIM class referenced by *ClassPath*.

2502 The set of properties to be included in the retrieved class shall be determined using the following  
 2503 algorithm:

- 2504 • Initially, the set of properties to be included is the set of properties exposed by the class to  
2505 be retrieved. This includes all the duplicates of any duplicate non-overridden properties.
- 2506 • If the *IncludedProperties* operation input parameter is supported by the WBEM protocol  
2507 and if its value is not NULL, it acts as a restricting filter on the properties to be included in  
2508 the returned class such that any properties exposed by the class to be retrieved that are  
2509 not named in that operation parameter are removed from the set of properties to be  
2510 included. Any duplicate or invalid property names in the *IncludedProperties* operation input  
2511 parameter shall be ignored. A non-NULL empty *IncludedProperties* list removes all  
2512 properties from the set of properties to be included.

2513 **Preconditions:**

- 2514 • The CIM class referenced by *ClassPath* shall exist in the namespace. If this is not satisfied, the  
2515 operation shall fail, indicating WIPG0214.

2516 **Postconditions:**

- 2517 • The CIM class shall have been returned as defined in the Description paragraph for this  
2518 operation.
- 2519 • Requirements on ACID properties:
  - 2520 – Atomicity: N/A
  - 2521 – Update Consistency: N/A
  - 2522 – Isolation: Required
  - 2523 – Durability: N/A

2524 **Error Messages:**

2525

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	

2526 **6.7.2 DeleteClass**2527 **Purpose:**

2528 Delete a CIM class.

2529 **Operation Input Parameters:**

2530

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

2531 **Operation Output Parameters:**

2532 None.

2533 **Description:**2534 The *DeleteClass* operation deletes the CIM class referenced by *ClassPath*.

2535

---

**2536 EXPERIMENTAL**

2537 If the WBEM protocol supports the *DeleteDependents* operation parameter, the following rules apply:

- 2538 • If *DeleteDependents* is TRUE, any classes that depend on the class referenced by  
2539 *ClassPath* in the way described below shall be deleted, and any instances of the class  
2540 referenced by *ClassPath* and of any classes depending on it shall be deleted according to  
2541 the rules defined for the *DeleteInstance* operation. If these rules cause the rejection of an  
2542 instance deletion, the *DeleteClass* operation shall fail.
- 2543 • If *DeleteDependents* is FALSE, the *DeleteClass* operation shall fail if any classes exist that  
2544 depend on the class referenced by *ClassPath* in the way described below, or if the class  
2545 referenced by *ClassPath* has any instances.

---

**2546 EXPERIMENTAL**

2547 If the WBEM protocol does not support the *DeleteDependents* operation parameter, the *DeleteClass*  
2548 operation shall fail if any classes exist that depend on the class referenced by *ClassPath* in the way  
2549 described below, or if the class referenced by *ClassPath* has any instances.

2550 For the purpose of the *DeleteClass* operation, the following classes are considered depending on the  
2551 class referenced by *ClassPath*:

- 2552 • Any subclasses of any class depending on the class referenced by *ClassPath*.
- 2553 • Any association classes referencing any class depending on the class referenced by  
2554 *ClassPath*.
- 2555 • Any classes defining a method with a parameter or a return value that is
  - 2556 – a reference to any class depending on the class referenced by *ClassPath*, or
  - 2557 – an embedded instance of any class depending on the class referenced by *ClassPath*,  
2558 or
  - 2559 – an embedded class depending on the class referenced by *ClassPath*.
- 2560 • Any classes defining a property that is
  - 2561 – an embedded instance of any class depending on the class referenced by *ClassPath*,  
2562 or
  - 2563 – an embedded class depending on the class referenced by *ClassPath*.

2564 Any classes or instances that are automatically deleted may reside in a different CIM namespace  
2565 (which may reside in a different WBEM server) than the class referenced by *ClassPath*.

2566 In case of error, the consistency requirements defined in [DSP0004](#) cannot be guaranteed, but should  
2567 be attempted to be satisfied in a best effort approach. In case of error, only a subset of the elements  
2568 to be deleted may have been deleted, but each element shall have either been deleted completely or  
2569 not at all. Also, classes shall only be deleted if all of its instances could be deleted successfully.

2570 NOTE: In a non-transactional implementation, this requires an order of deletion that starts with those elements  
2571 that do not depend on the deletion of other elements.

**2572 Preconditions:**

- 2573 • The CIM class referenced by *ClassPath* shall exist in the namespace. If this is not satisfied, the  
2574 operation shall fail, indicating WIPG0214.

2575 **Postconditions:**

- 2576
- The CIM class referenced by *ClassPath* shall have been deleted.
- 2577
- If *DeleteDependents* was TRUE:
    - 2578 – any dependent classes and instances shall have been deleted as defined in the
    - 2579 Description paragraph for this operation, and
    - 2580 – any management profile defined implicit deletions of other CIM instances shall have
    - 2581 happened, and
    - 2582 – any management profile defined effects of the deletion of all of these CIM instances
    - 2583 on any underlying resources shall have happened.
  - The consistency requirements defined in [DSP0004](#) shall be satisfied for any classes and instances related to the deleted classes and instances.
  - Requirements on ACID properties:
    - 2587 – Atomicity: Required, if dependent classes and instances are handled by rejection, as
    - 2588 defined in 5.8.9. Recommended, if dependent classes and instances are handled by
    - 2589 delete propagation, as defined in 5.8.9.
    - 2590 – Update Consistency: Required, if dependent classes and instances are handled by
    - 2591 rejection, as defined in 5.8.9. Recommended, if dependent classes and instances are
    - 2592 handled by delete propagation, as defined in 5.8.9.
    - 2593 – Isolation: Required, if dependent classes and instances are handled by rejection, as
    - 2594 defined in 5.8.9. Recommended, if dependent classes and instances are handled by
    - 2595 delete propagation, as defined in 5.8.9.
    - 2596 – Durability: Required

2597 **Error Messages:**

2598

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.	

2599 **6.7.3 ModifyClass**

2600 **Purpose:**

2601 Change the definition of a CIM class.

2602 **Operation Input Parameters:**

2603

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

2604 **Operation Output Parameters:**

2605 None.

2606 **Description:**

2607 The *ModifyClass* operation changes the definition of the CIM class referenced by *ClassPath*.

2608 Within the restrictions specified in the preconditions, the definition of the class referenced by  
2609 *ClassPath* is replaced with the definition specified in *ModifiedClass*, as follows:

- 2610 • Any elements previously defined in the class to be changed that are not specified in  
2611 *ModifiedClass* shall be removed from the class to be changed.
- 2612 • Any elements previously defined in the class to be changed that are also specified in  
2613 *ModifiedClass* shall be replaced with the definition from *ModifiedClass*.
- 2614 • Any elements not previously defined in the class to be changed that are specified in  
2615 *ModifiedClass* shall be added to the class to be changed, as defined in *ModifiedClass*.

2616 Any instances whose creation class is the class referenced by *ClassPath* or one of its subclasses  
2617 shall be changed to reflect the changes to the class, as follows:

- 2618 • Added properties are reflected using the rules defined in the *ModifyInstance* operation  
2619 when processing a list of these new properties with their values set to their class defined  
2620 default values, or NULL where no class defined default value is defined.

2621 Any other changes to the class that are compatible with the preconditions do not affect existing  
2622 instances, for the following reasons:

- 2623 • A compatible removal of properties from a class can only happen for overridden properties  
2624 or for properties that move to a superclass, both of which is equivalent to potential changes

2625 of qualifier values and the default property value. Changes of qualifier values do not affect  
2626 instances. A changed default value only affects new instances, but not existing instances.

2627 • A compatible change of existing property definitions can only include potential changes of  
2628 qualifier values and the default property value. Changes of qualifier values do not affect  
2629 instances. A changed default value only affects new instances, but not existing instances.

2630 • A compatible change of values of class qualifiers does not affect instances of the class.

2631 • A compatible change to a method definition does not affect instances of the class.

#### 2632 **Preconditions:**

2633 • The CIM class referenced by *ClassPath* shall exist in the namespace. If this is not satisfied, the  
2634 operation shall fail, indicating WIPG0214.

2635 • The name of the class defined by *ModifiedClass* shall be the name of the class referenced by  
2636 *ClassPath*. If this is not satisfied, the operation shall fail, indicating WIPG0208.

2637 • If the class referenced by *ClassPath* has a superclass, the class defined by *ModifiedClass* shall  
2638 specify a superclass with the same name as that superclass. If the class referenced by  
2639 *ClassPath* has no superclass, the class defined by *ModifiedClass* shall not specify a superclass.  
2640 If this is not satisfied, the operation shall fail, indicating WIPG0226.

2641 • The class defined by *ModifiedClass* shall only specify elements that when applied to the class to  
2642 be modified, result in a class definition that satisfies any consistency and backward compatibility  
2643 requirements defined in [DSP0004](#). For example, qualifiers with flavor *DisableOverride* shall not  
2644 be overridden, or data types of overridden properties shall not be changed. If this is not  
2645 satisfied, the operation shall fail, indicating WIPG0231.

#### 2646 **Postconditions:**

2647 • The definition of the class referenced by *ClassPath* shall have been modified as defined in the  
2648 Description paragraph for this operation.

2649 • Any instances of the class or its subclasses shall have been changed as defined in the  
2650 Description paragraph for this operation.

2651 • The consistency and backward compatibility requirements defined in [DSP0004](#) shall be satisfied  
2652 for the modified class.

2653 • Requirements on ACID properties:

2654 – Atomicity: Required

2655 – Update Consistency: Required

2656 – Isolation: Required

2657 – Durability: Required

#### 2658 **Error Messages:**

2659

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	

2660 **6.7.4 CreateClass**

2661 **Purpose:**

2662 Create a CIM class.

2663 **Operation Input Parameters:**

2664

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

2665 **Operation Output Parameters:**

2666

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

2667 **Description:**

2668 The *CreateClass* operation creates a CIM class in the namespace referenced by *NamespacePath*,  
2669 using the class definition specified in *NewClass*, and returns the class path of the new class.

2670 If properties or methods defined in *NewClass* are intended to override properties or methods defined  
2671 in a superclass of *NewClass*, then they shall define an *OVERRIDE* qualifier in their definition in  
2672 *NewClass*. The *CreateClass* operation shall not add such qualifiers automatically.

2673 **Preconditions:**

- 2674       • The CIM namespace referenced by *NamespacePath* shall exist. If this is not satisfied, the  
2675       operation shall fail, indicating WIPG0204.
- 2676       • The CIM class to be created shall not exist in the namespace referenced by *NamespacePath*. If  
2677       this is not satisfied, the operation shall fail, indicating WIPG0217.
- 2678       • If *NewClass* specifies a superclass, that superclass shall exist in the namespace referenced by  
2679       *NamespacePath*. If this is not satisfied, the operation shall fail, indicating WIPG0226.
- 2680       NOTE: [DSP0004](#) does not provide for inheritance relationships that cross namespace boundaries.
- 2681       • The definition of *NewClass* shall satisfy any consistency requirements defined in [DSP0004](#). If  
2682       this is not satisfied, the operation shall fail, indicating WIPG0208.

2683 **Postconditions:**

- 2684       • The CIM class shall have been created as defined in the Description paragraph for this  
2685       operation.
- 2686       • Requirements on ACID properties:
- 2687       – Atomicity: Required
- 2688       – Update Consistency: Required
- 2689       – Isolation: Required
- 2690       – Durability: Required

2691 **Error Messages:**

2692

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	

2693 **6.8 Class enumeration operations**

2694 This subclause defines class enumeration operations (operations that enumerate CIM classes and return  
2695 those classes or their class paths).

2696 **6.8.1 GetTopClassesWithPath**

2697 **Purpose:**

2698 Enumerate all top classes (i.e., classes that have no superclasses) in a namespace and return these  
2699 classes together with their paths.

2700 **Operation Input Parameters:**

2701

Generic Name	Generic Type	Requirement	Description
NamespacePath	NamespacePath	Mandatory	Namespace path of the CIM namespace the enumeration is executed on (Context Parameter)
IncludeSubclasses	boolean	Mandatory	Indicates whether the entire tree of subclasses of the top classes is to be included in the result set or just the top 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

2702 **Operation Output Parameters:**

2703

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

2704 **Description:**

2705 The *GetTopClassesWithPath* operation enumerates all CIM classes in the namespace specified in  
2706 *NamespacePath* that do not have a superclass defined, and returns these CIM classes together with  
2707 their class paths.

2708 The consistency model defined in 5.8 applies.

2709 If *IncludeSubclasses* is TRUE, then the set of returned classes shall consist of all classes that exist  
2710 in the namespace referenced by *NamespacePath*. Otherwise, the set of returned classes shall  
2711 consist of those classes that exist in the namespace referenced by *NamespacePath* and do not have  
2712 a superclass defined. In both cases, this includes any association or indication classes.

2713 Note that unlike the *GetSubClassesWithPath* operation (see 6.8.3), this operation intentionally does  
2714 not have an *IncludeInheritedElements* parameter, because there is no specified class that would be  
2715 the basis for that parameter.

2716 **Preconditions:**

- 2717       • The CIM namespace referenced by *NamespacePath* shall exist. If this is not satisfied, the  
2718       operation shall fail, indicating WIPG0204.

2719 **Postconditions:**

- 2720       • The top classes with their class paths shall have been returned as defined in the Description  
2721       paragraph for this operation.
- 2722       • Requirements on ACID properties:
- 2723       – Atomicity: N/A
- 2724       – Update Consistency: N/A
- 2725       – Isolation: Required at the level of single classes, as defined in 5.8.
- 2726       – Durability: N/A

2727 **Error Messages:**

2728

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	

2729 **6.8.2 GetTopClassPaths**

2730 **Purpose:**

2731 Enumerate all top classes (i.e., classes that have no superclasses) in a namespace and return their  
 2732 class paths.

2733 **Operation Input Parameters:**

2734

Generic Name	Generic Type	Requirement	Description
NamespacePath	NamespacePath	Mandatory	Namespace path of the CIM namespace the enumeration is executed on (Context Parameter)
IncludeSubclasses	boolean	Mandatory	Indicates whether the entire tree of subclasses of the top classes are to be included in the result set or just the top classes

2735 **Operation Output Parameters:**

2736

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

2737 **Description:**

2738 The *GetTopClassPaths* operation enumerates all CIM classes in the namespace specified in  
 2739 *NamespacePath* that do not have a superclass defined, and returns the class paths of these classes.

2740 The consistency model defined in 5.8 applies.

2741 If *IncludeSubclasses* is TRUE, then the set of returned classes shall consist of all classes that exist  
 2742 in the namespace referenced by *NamespacePath*. Otherwise, the set of returned classes shall  
 2743 consist of those classes that exist in the namespace referenced by *NamespacePath* and do not have  
 2744 a superclass defined. In both cases, this includes any association or indication classes.

2745 **Preconditions:**

- 2746 • The CIM namespace referenced by *NamespacePath* shall exist. If this is not satisfied, the  
 2747 operation shall fail, indicating WIPG0204.

2748 **Postconditions:**

- 2749 • The class paths of the top classes shall have been returned as defined in the Description  
 2750 paragraph for this operation.
- 2751 • Requirements on ACID properties:
  - 2752 – Atomicity: N/A
  - 2753 – Update Consistency: N/A
  - 2754 – Isolation: Required at the level of single classes, as defined in 5.8.
  - 2755 – Durability: N/A

2756  
2757**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	
WIPG0243	Timeout	Optional	Infrastructure	
WIPG0227	Other failure	Optional	Infrastructure	

2758 **6.8.3 GetSubClassesWithPath**2759 **Purpose:**

2760 Enumerate the subclasses of a class and return these classes together with their class paths.

2761 **Operation Input Parameters:**  
2762

Generic Name	Generic Type	Requirement	Description
ClassPath	ClassPath	Mandatory	Class path of the CIM class the subclasses of which are to be enumerated (Context Parameter)
IncludeSubclasses	boolean	Mandatory	Indicates whether the entire tree of subclasses of the given class is to be included in the result set or just one level
IncludeInherited-Elements	boolean	Mandatory	Indicates whether any elements inherited from superclasses 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.

2763 **Operation Output Parameters:**  
2764

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

2765 **Description:**

2766 The *GetSubClassesWithPath* operation enumerates all subclasses of the class referenced by  
2767 *ClassPath* and returns these CIM classes together with their class paths.

2768 The consistency model defined in 5.8 applies.

2769 If *IncludeSubclasses* is TRUE, then the set of returned classes shall consist of all direct and indirect  
2770 subclasses of the class referenced by *ClassPath*. Otherwise, the set of returned classes shall consist  
2771 only of all direct subclasses of the class referenced by *ClassPath*. In both cases, this includes any  
2772 association or indication classes.

2773 If *IncludeInheritedElements* is TRUE, then the set of CIM elements in each returned class shall  
2774 consist of all elements exposed by that class. Otherwise, the set of CIM elements in each returned  
2775 class shall consist only of all elements defined in the class referenced by *ClassPath* (including  
2776 overriding elements). This is also known as reducing the elements to *local-only* elements..

2777 **Preconditions:**

- 2778 • The CIM class referenced by *ClassPath* shall exist in the namespace. If this is not satisfied, the  
2779 operation shall fail, indicating WIPG0214.

2780 **Postconditions:**

- 2781 • The subclasses with their class paths shall have been returned as defined in the Description  
2782 paragraph for this operation.
- 2783 • Requirements on ACID properties:
  - 2784 – Atomicity: N/A
  - 2785 – Update Consistency: N/A
  - 2786 – Isolation: Required at the level of single classes, as defined in 5.8.
  - 2787 – Durability: N/A

2788 **Error Messages:**  
2789

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	

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	
WIPG0214	Class not found	Mandatory	Infrastructure	
WIPG0243	Timeout	Optional	Infrastructure	
WIPG0227	Other failure	Optional	Infrastructure	

#### 2790 6.8.4 GetSubClassPaths

##### 2791 Purpose:

2792 Enumerate the subclasses of a class and return their class paths.

##### 2793 Operation Input Parameters:

2794

Generic Name	Generic Type	Requirement	Description
ClassPath	ClassPath	Mandatory	Class path of the CIM class the subclasses of which are to be enumerated (Context Parameter)
IncludeSubclasses	boolean	Mandatory	Indicates whether the entire tree of subclasses of the given class is to be included in the result set or just one level

##### 2795 Operation Output Parameters:

2796

Generic Name	Generic Type	Requirement	Description
ClassPathList	ClassPath [ ]	Mandatory	Sequence of class paths of the enumerated subclasses

##### 2797 Description:

2798 The *GetSubClassPaths* operation enumerates all subclasses of the class referenced by *ClassPath*  
2799 and returns the addresses of these CIM classes.

2800 The consistency model defined in 5.8 applies.

2801 If *IncludeSubclasses* is TRUE, then the set of returned classes shall consist of all direct and indirect  
2802 subclasses of the class referenced by *ClassPath*. Otherwise, the set of returned classes shall consist  
2803 only of all direct subclasses of the class referenced by *ClassPath*. In both cases, this includes any  
2804 association or indication classes.

##### 2805 Preconditions:

- 2806 • The CIM class referenced by *ClassPath* shall exist in the namespace. If this is not satisfied, the  
2807 operation shall fail, indicating WIPG0214.

2808 **Postconditions:**

- 2809       • The class path of the subclasses shall have been returned as defined in the Description
- 2810        paragraph for this operation.
- 2811       • Requirements on ACID properties:
- 2812        – Atomicity: N/A
- 2813        – Update Consistency: N/A
- 2814        – Isolation: Required at the level of single classes, as defined in 5.8.
- 2815        – Durability: N/A

2816 **Error Messages:**

2817

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	Invalid input parameter value	Mandatory	Infrastructure	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure	
WIPG0214	Class not found	Mandatory	Infrastructure	
WIPG0243	Timeout	Optional	Infrastructure	
WIPG0227	Other failure	Optional	Infrastructure	

2818 **6.8.5 GetAssociatedClassesWithPath**2819 **Purpose:**

2820 Enumerate the classes that are associated with a given source class and return those classes  
 2821 together with their class paths.

2822 **Operation Input Parameters:**  
2823

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

2824 **Operation Output Parameters:**  
2825

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

2826 **Description:**

2827 The *GetAssociatedClassesWithPath* operation traverses an association from a class on a starting  
 2828 end to classes on all of its far ends and returns the associated CIM classes together with their class  
 2829 paths.

- 2830 The set of associated classes to be returned shall be determined using the following algorithm:
- 2831 • Initially, the set of classes to be returned is the set of all classes associated to any of the  
2832 far ends of all associations referencing the starting class.
  - 2833 • If the *AssociationClassName* operation input parameter is not NULL, it acts as a restricting  
2834 filter on the classes to be returned such that each class that is associated with the starting  
2835 class using an association class where the class or one of its superclasses does not have  
2836 the name specified in *AssociationClassName*, is removed from the set of classes to be  
2837 returned. There shall be no validity checking performed for the *AssociationClassName*  
2838 operation input parameter.
  - 2839 • If the *AssociatedClassName* operation input parameter is not NULL, it acts as a restricting  
2840 filter on the classes to be returned such that each class where the class or one of its  
2841 superclasses does not have the name specified in *AssociatedClassName*, is removed from  
2842 the set of classes to be returned. There shall be no validity checking performed for the  
2843 *AssociatedClassName* operation input parameter.
  - 2844 • If the *RoleName* operation input parameter is not NULL, it acts as a restricting filter on the  
2845 classes to be returned such that each class that is associated with the starting class using  
2846 an association class that has a role name on its starting end that is not the role name  
2847 specified in *RoleName*, is removed from the set of classes to be returned. There shall be  
2848 no validity checking performed for the *RoleName* operation input parameter.
  - 2849 • If the *AssociatedRoleName* operation input parameter is not NULL, it acts as a restricting  
2850 filter on the classes to be returned such that each class that is associated with the starting  
2851 class using an association class that has a role name on the far end referencing that class  
2852 that is not the role name specified in *AssociatedRoleName*, is removed from the set of  
2853 classes to be returned. There shall be no validity checking performed for the  
2854 *AssociatedRoleName* operation input parameter.
- 2855 The consistency model defined in 5.8 applies.
- 2856 The set of properties to be included in each returned associated class shall be determined using the  
2857 following algorithm:
- 2858 • Initially, the set of properties to be included is the set of properties exposed by the class.  
2859 This includes all the duplicates of any duplicate non-overridden properties.
  - 2860 • If the *IncludedProperties* operation input parameter is supported by the WBEM protocol  
2861 and if its value is not NULL, it acts as a restricting filter on the properties to be included in  
2862 the returned classes such that any properties exposed by the associated class that are not  
2863 named in that operation parameter are removed from the set of properties to be included.  
2864 Any duplicate or invalid property names in the *IncludedProperties* operation input  
2865 parameter shall be ignored. A non-NULL empty *IncludedProperties* list removes all  
2866 properties from the set of properties to be included.
- 2867 **Preconditions:**
- 2868 • The CIM class referenced by *ClassPath* shall exist in the namespace. If this is not satisfied, the  
2869 operation shall fail, indicating WIPG0214.
  - 2870 • The *IncludedProperties* operation parameter, if supported by the WBEM protocol, shall only be  
2871 specified with a non-NULL value if the *AssociatedClassName* operation input parameter is also  
2872 non-NULL. If this is not satisfied, the operation shall fail, indicating WIPG0208.
- 2873 NOTE: Specifying a non-NULL value for *AssociatedClassName* ensures that the associated classes have the  
2874 class specified in *AssociatedClassName* as a common superclass.

2875 **Postconditions:**

- 2876       • The associated classes with their class paths shall have been returned as described in the  
2877       Description paragraph for this operation.
- 2878       • Requirements on ACID properties:
- 2879       – Atomicity: N/A
- 2880       – Update Consistency: N/A
- 2881       – Isolation: Required at the level of single classes, as defined in 5.8.
- 2882       – Durability: N/A

2883 **Error Messages:**

2884

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	

2885 **6.8.6 GetAssociatedClassPaths**

2886 **Purpose:**

2887 Enumerate the classes that are associated with a given source class and return their class paths.

2888 **Operation Input Parameters:**

2889

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

2890 **Operation Output Parameters:**

2891

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

2892 **Description:**

2893 The *GetAssociatedClassPaths* operation traverses an association from a class on a starting end to  
2894 classes on all of its far ends and returns the class paths of the associated CIM classes.

2895 The set of associated classes to be returned shall be determined using the following algorithm:

- 2896 • Initially, the set of classes to be returned is the set of all classes associated to any of the  
2897 far ends of all associations referencing the starting class.
- 2898 • If the *AssociationClassName* operation input parameter is not NULL, it acts as a restricting  
2899 filter on the classes to be returned such that each class that is associated with the starting  
2900 class using an association class where the class or one of its superclasses does not have  
2901 the name specified in *AssociationClassName*, is removed from the set of classes to be  
2902 returned. There shall be no validity checking performed for the *AssociationClassName*  
2903 operation input parameter.
- 2904 • If the *AssociatedClassName* operation input parameter is not NULL, it acts as a restricting  
2905 filter on the classes to be returned such that each class where the class or one of its  
2906 superclasses does not have the name specified in *AssociatedClassName*, is removed from  
2907 the set of classes to be returned. There shall be no validity checking performed for the  
2908 *AssociatedClassName* operation input parameter.

2909           • If the *RoleName* operation input parameter is not NULL, it acts as a restricting filter on the  
 2910 classes to be returned such that each class that is associated with the starting class using  
 2911 an association class that has a role name on its starting end that is not the role name  
 2912 specified in *RoleName*, is removed from the set of classes to be returned. There shall be  
 2913 no validity checking performed for the *RoleName* operation input parameter.

2914           • If the *AssociatedRoleName* operation input parameter is not NULL, it acts as a restricting  
 2915 filter on the classes to be returned such that each class that is associated with the starting  
 2916 class using an association class that has a role name on the far end referencing that class  
 2917 that is not the role name specified in *AssociatedRoleName*, is removed from the set of  
 2918 classes to be returned. There shall be no validity checking performed for the  
 2919 *AssociatedRoleName* operation input parameter.

2920           The consistency model defined in 5.8 applies.

2921   **Preconditions:**

2922           • The CIM class referenced by *ClassPath* shall exist in the namespace. If this is not satisfied, the  
 2923 operation shall fail, indicating WIPG0214.

2924   **Postconditions:**

2925           • The class paths of the associated classes shall have been returned as described in the  
 2926 Description paragraph for this operation.

2927           • Requirements on ACID properties:

2928           – Atomicity: N/A

2929           – Update Consistency: N/A

2930           – Isolation: Required at the level of single classes, as defined in 5.8.

2931           – Durability: N/A

2932   **Error Messages:**

2933

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	

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0243	Timeout	Optional	Infrastructure	
WIPG0227	Other failure	Optional	Infrastructure	

2934 **6.8.7 GetReferencingClassesWithPath**

2935 **Purpose:**

2936 Enumerate the association classes that reference a given source class and return these classes  
 2937 together with their class paths.

2938 **Operation Input Parameters:**  
 2939

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

2940 **Operation Output Parameters:**  
 2941

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

2942 **Description:**

2943 The *GetReferencingClassesWithPath* operation traverses an association from a class on a starting  
2944 end to classes on all of its far ends and returns the CIM association classes traversed together with  
2945 their class paths.

2946 The set of association classes to be returned shall be determined using the following algorithm:

- 2947 • Initially, the set of classes to be returned is the set of all association classes referencing  
2948 the starting class.
- 2949 • If the *AssociationClassName* operation input parameter is not NULL, it acts as a restricting  
2950 filter on the classes to be returned such that each association class where the class or one  
2951 of its superclasses does not have the name specified in *AssociationClassName*, is  
2952 removed from the set of classes to be returned. There shall be no validity checking  
2953 performed for the *AssociationClassName* operation input parameter.
- 2954 • If the *AssociatedClassName* operation input parameter is not NULL, it acts as a restricting  
2955 filter on the classes to be returned such that each association class that has a set of  
2956 references on its far ends such that none of these classes or their superclasses have the  
2957 name specified in *AssociatedClassName*, is removed from the set of classes to be  
2958 returned. There shall be no validity checking performed for the *AssociatedClassName*  
2959 operation input parameter.
- 2960 • If the *RoleName* operation input parameter is not NULL, it acts as a restricting filter on the  
2961 classes to be returned such that each association class that has a role name on its starting  
2962 end that is not the role name specified in *RoleName*, is removed from the set of classes to  
2963 be returned. There shall be no validity checking performed for the *RoleName* operation  
2964 input parameter.
- 2965 • If the *AssociatedRoleName* operation input parameter is not NULL, it acts as a restricting  
2966 filter on the classes to be returned such that each association class that has a set of role  
2967 names on its far ends such that none of them is the role name specified in  
2968 *AssociatedRoleName*, is removed from the set of classes to be returned. There shall be no  
2969 validity checking performed for the *AssociatedRoleName* operation input parameter.

2970 The consistency model defined in 5.8 applies.

2971 The set of properties to be included in each returned association class shall be determined using the  
2972 following algorithm:

- 2973 • Initially, the set of properties to be included is the set of properties exposed by the  
2974 association class. This includes all the duplicates of any duplicate non-overridden  
2975 properties.
- 2976 • If the *IncludedProperties* operation input parameter is supported by the WBEM protocol  
2977 and if its value is not NULL, it acts as a restricting filter on the properties to be included in  
2978 the returned classes such that any properties exposed by the associated class that are not  
2979 named in that operation parameter are removed from the set of properties to be included.  
2980 Any duplicate or invalid property names in the *IncludedProperties* operation input  
2981 parameter shall be ignored. A non-NULL empty *IncludedProperties* list removes all  
2982 properties from the set of properties to be included.

2983 **Preconditions:**

- 2984 • The CIM class referenced by *ClassPath* shall exist in the namespace. If this is not satisfied, the  
2985 operation shall fail, indicating WIPG0214.
- 2986 • The *IncludedProperties* operation parameter, if supported by the WBEM protocol, shall only be  
2987 specified with a non-NULL value if the *AssociationClassName* operation input parameter is also  
2988 non-NULL. If this is not satisfied, the operation shall fail, indicating WIPG0208.

2989 NOTE: Specifying a non-NULL value for *AssociationClassName* ensures that the association classes have the  
2990 class specified in *AssociationClassName* as a common superclass.

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 **6.8.8 GetReferencingClassPaths**

3002 **Purpose:**

3003 Enumerate the association classes that reference a given source class and return their class paths.

3004 **Operation Input Parameters:**  
3005

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

3006 **Operation Output Parameters:**  
3007

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

3008 **Description:**

3009 The *GetReferencingClassPaths* operation traverses an association from a class on a starting end to  
3010 classes on all of its far ends and returns the class paths of the CIM association classes traversed.

3011 The set of association classes to be returned shall be determined using the following algorithm:

- 3012 • Initially, the set of classes to be returned is the set of all association classes referencing  
3013 the starting class.
- 3014 • If the *AssociationClassName* operation input parameter is not NULL, it acts as a restricting  
3015 filter on the classes to be returned such that each association class where the class or one  
3016 of its superclasses does not have the name specified in *AssociationClassName*, is  
3017 removed from the set of classes to be returned. There shall be no validity checking  
3018 performed for the *AssociationClassName* operation input parameter.
- 3019 • If the *AssociatedClassName* operation input parameter is not NULL, it acts as a restricting  
3020 filter on the classes to be returned such that each association class that has a set of  
3021 references on its far ends such that none of these classes or their superclasses have the  
3022 name specified in *AssociatedClassName*, is removed from the set of classes to be  
3023 returned. There shall be no validity checking performed for the *AssociatedClassName*  
3024 operation input parameter.
- 3025 • If the *RoleName* operation input parameter is not NULL, it acts as a restricting filter on the  
3026 classes to be returned such that each association class that has a role name on its starting  
3027 end that is not the role name specified in *RoleName*, is removed from the set of classes to  
3028 be returned. There shall be no validity checking performed for the *RoleName* operation  
3029 input parameter.

- 3030           • If the *AssociatedRoleName* operation input parameter is not NULL, it acts as a restricting  
 3031 filter on the classes to be returned such that each association class that has a set of role  
 3032 names on its far ends such that none of them is the role name specified in  
 3033 *AssociatedRoleName*, is removed from the set of classes to be returned. There shall be no  
 3034 validity checking performed for the *AssociatedRoleName* operation input parameter.

3035           The consistency model defined in 5.8 applies.

3036 **Preconditions:**

- 3037           • The CIM class referenced by *ClassPath* shall exist in the namespace. If this is not satisfied, the  
 3038 operation shall fail, indicating WIPG0214.

3039 **Postconditions:**

- 3040           • The association classes with their class paths shall have been returned as described in the  
 3041 Description paragraph for this operation.
- 3042           • Requirements on ACID properties:
- 3043           – Atomicity: N/A
- 3044           – Update Consistency: N/A
- 3045           – Isolation: Required at the level of single classes, as defined in 5.8.
- 3046           – Durability: N/A

3047 **Error Messages:**

3048

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	

3049 **6.9 Qualifier type operations**

3050 This subclause defines operations that deal with qualifier types. As defined in [DSP0004](#), qualifier types  
 3051 represent the declarations of qualifiers, not their values.

3052 **6.9.1 GetQualifierType**3053 **Purpose:**

3054 Retrieve a qualifier type.

3055 **Operation Input Parameters:**

3056

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

3057 **Operation Output Parameters:**

3058

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

3059 **Description:**3060 The *GetQualifierType* operation retrieves the CIM qualifier type referenced by *QualifierTypePath*.3061 **Preconditions:**

- 3062 • The CIM qualifier type referenced by *QualifierTypePath* shall exist in the namespace. If this is  
3063 not satisfied, the operation shall fail, indicating WIPG0215.

3064 **Postconditions:**

- 3065 • The qualifier type shall have been returned as described in the Description paragraph for this  
3066 operation.
- 3067 • Requirements on ACID properties:
- 3068 – Atomicity: N/A
- 3069 – Update Consistency: N/A
- 3070 – Isolation: Required
- 3071 – Durability: N/A

3072 **Error Messages:**

3073

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	

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	
WIPG0215	Qualifier type not found	Mandatory	Infrastructure	
WIPG0243	Timeout	Optional	Infrastructure	
WIPG0227	Other failure	Optional	Infrastructure	

3074 **6.9.2 DeleteQualifierType**

3075 **Purpose:**

3076 Delete a qualifier type.

3077 **Operation Input Parameters:**

3078

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

3079 **Operation Output Parameters:**

3080 None.

3081 **Description:**

3082 The *DeleteQualifierType* operation deletes the CIM qualifier type referenced by *QualifierTypePath*.

3083 As defined in [DSP0004](#), any namespace needs to contain qualifier types for the meta qualifiers and  
 3084 standard qualifiers, and may contain qualifier types for the optional qualifiers. Thus, deleting any  
 3085 required qualifier types from a namespace will render that namespace non-compliant to [DSP0004](#).

3086 **Preconditions:**

- 3087 • The CIM qualifier type referenced by *QualifierTypePath* shall exist in the namespace. If this is  
 3088 not satisfied, the operation shall fail, indicating WIPG0215.
- 3089 • The qualifier identified by *QualifierTypePath* shall not be specified on any element in the same  
 3090 namespace. If this is not satisfied, the operation shall fail, indicating WIPG0233.

3091 **Postconditions:**

- 3092 • The CIM qualifier type shall have been deleted as described in the Description paragraph for  
 3093 this operation.
- 3094 • Requirements on ACID properties:  
 3095 – Atomicity: Required

- 3096 – Update Consistency: Required
- 3097 – Isolation: Required
- 3098 – Durability: Required

3099 **Error Messages:**

3100

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	
WIPG0233	Qualifier type is used	Mandatory	Infrastructure	
WIPG0243	Timeout	Optional	Infrastructure	
WIPG0227	Other failure	Optional	Infrastructure	

3101 **6.9.3 ModifyQualifierType**

3102 **Purpose:**

3103 Change the definition of a CIM qualifier type.

3104 **Operation Input Parameters:**

3105

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

3106 **Operation Output Parameters:**

3107 None.

3108 **Description:**

3109 The *ModifyQualifierType* operation changes the definition of the CIM qualifier type referenced by  
 3110 *QualifierTypePath*.

3111 The definition of the qualifier type referenced by *QualifierTypePath* is replaced with the definition  
 3112 specified in *ModifiedQualifierType*.

3113 As defined in [DSP0004](#), any namespace needs to contain qualifier types for the meta qualifiers and  
 3114 standard qualifiers, and may contain qualifier types for the optional qualifiers. Thus, changing these  
 3115 qualifier types in a namespace inconsistently with their [DSP0004](#) definition will render that  
 3116 namespace non-compliant to [DSP0004](#).

3117 **Preconditions:**

- 3118 • The CIM qualifier type referenced by *QualifierTypePath* shall exist in the namespace. If this is  
 3119 not satisfied, the operation shall fail, indicating WIPG0215.
- 3120 • The name of the qualifier type defined by *ModifiedQualifierType* shall be the name of the  
 3121 qualifier type referenced by *QualifierTypePath*. If this is not satisfied, the operation shall fail,  
 3122 indicating WIPG0208.
- 3123 • The request to modify the qualifier type shall satisfy any backward compatibility requirements  
 3124 defined in [DSP0004](#). If this is not satisfied, the operation shall fail, indicating WIPG0234.
- 3125 • If the qualifier type referenced by *QualifierTypePath* is one of the qualifiers defined in [DSP0004](#),  
 3126 (i.e., meta, standard, and optional qualifiers), the new definition of the qualifier in  
 3127 *ModifiedQualifierType* shall be consistent with the definition of the qualifier in [DSP0004](#). If this is  
 3128 not satisfied, the operation shall fail, indicating WIPG0245.

3129 **Postconditions:**

- 3130 • The definition of the qualifier type referenced by *QualifierTypePath* shall have been modified as  
 3131 defined in the Description paragraph for this operation.
- 3132 • The backward compatibility requirements defined in [DSP0004](#) shall be satisfied for the modified  
 3133 qualifier type.
- 3134 • Requirements on ACID properties:
  - 3135 – Atomicity: Required
  - 3136 – Update Consistency: Required
  - 3137 – Isolation: Required
  - 3138 – Durability: Required

3139 **Error Messages:**

3140

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	

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	
WIPG0215	Qualifier type not found	Mandatory	Infrastructure	
WIPG0234	Incompatible modification of qualifier type	Mandatory	Infrastructure	
WIPG0245	Qualifier type inconsistent with <a href="#">DSP0004</a>	Mandatory	Infrastructure	
WIPG0243	Timeout	Optional	Infrastructure	
WIPG0227	Other failure	Optional	Infrastructure	

#### 3141 **6.9.4 CreateQualifierType**

##### 3142 **Purpose:**

3143 Create a CIM qualifier type.

##### 3144 **Operation Input Parameters:**

3145

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

##### 3146 **Operation Output Parameters:**

3147

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

##### 3148 **Description:**

3149 The *CreateQualifierType* operation creates a CIM qualifier type in the namespace referenced by  
3150 *NamespacePath*, using the qualifier type definition specified in *NewQualifierType*, and returns the  
3151 qualifier type path of the new qualifier type.

3152 As defined in [DSP0004](#), any namespace needs to contain qualifier types for the meta qualifiers and  
3153 standard qualifiers, and may contain qualifier types for the optional qualifiers. Thus, creating these  
3154 qualifier types in a namespace inconsistently with their [DSP0004](#) definition will render that  
3155 namespace non-compliant to [DSP0004](#).

3156 **Preconditions:**

- 3157 • The CIM namespace referenced by *NamespacePath* shall exist. If this is not satisfied, the  
3158 operation shall fail, indicating WIPG0204.
- 3159 • The CIM qualifier type to be created shall not exist in the namespace referenced by  
3160 *NamespacePath*. If this is not satisfied, the operation shall fail, indicating WIPG0208.
- 3161 • If the qualifier type defined in *NewQualifierType* is one of the qualifiers defined in [DSP0004](#),  
3162 (i.e., meta, standard, and optional qualifiers), the definition of the qualifier in *NewQualifierType*  
3163 shall be consistent with the definition of the qualifier in [DSP0004](#). If this is not satisfied, the  
3164 operation shall fail, indicating WIPG0245.

3165 **Postconditions:**

- 3166 • The CIM qualifier type shall have been created as defined in the Description paragraph for this  
3167 operation.
- 3168 • Requirements on ACID properties:
  - 3169 – Atomicity: Required
  - 3170 – Update Consistency: Required
  - 3171 – Isolation: Required
  - 3172 – Durability: Required

3173 **Error Messages:**

3174

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 <a href="#">DSP0004</a>	Mandatory	Infrastructure	
WIPG0243	Timeout	Optional	Infrastructure	
WIPG0227	Other failure	Optional	Infrastructure	

3175 **6.9.5 EnumerateQualifierTypesWithPath**3176 **Purpose:**

3177 Enumerate the qualifier types in a namespace.

3178 **Operation Input Parameters:**

3179

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)

3180 **Operation Output Parameters:**

3181

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

3182 **Description:**

3183 The *EnumerateQualifierTypesWithPath* operation enumerates all CIM qualifier types in the  
 3184 namespace referenced by *NamespacePath*, and returns these qualifier types together with their  
 3185 qualifier type paths.

3186 **Preconditions:**

- 3187
- 3188 • The CIM namespace referenced by *NamespacePath* shall exist. If this is not satisfied, the operation shall fail, indicating WIPG0204.

3189 **Postconditions:**

- 3190
- 3191 • The CIM qualifier types with their qualifier type paths shall have been enumerated as defined in the Description paragraph for this operation.
  - 3192 • Requirements on ACID properties:
    - 3193 – Atomicity: N/A
    - 3194 – Update Consistency: N/A
    - 3195 – Isolation: Required at the level of single qualifier types, as defined in 5.8.
    - 3196 – Durability: N/A

3197 **Error Messages:**

3198

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	

Message ID	Message Name	Requirement	Sources	Additional Description
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

3203 This annex provides ideas for future operations or extensions to existing operations.

#### 3204 **A.1 Test for property modifiability**

3205 Today, management profiles specify the modifiability of properties or an algorithm how to find out their  
3206 modifiability at runtime. Usually, this includes the overhead of capability based mechanisms most of the  
3207 time at the level of single properties. Because of this overhead, it is defined rarely in profiles and thus left  
3208 to be decided by the implementation, with no defined way for a client to find out about it upfront.

3209 An operation (or an extension to an existing operation) that allows testing for modifiability of properties in  
3210 a consistent way without depending on hard wired understanding of profile defined modifiability or profile  
3211 defined algorithms to find out modifiability would be a worthwhile extension.

#### 3212 **A.2 Retrieval of associated instance graph**

3213 Today, a graph of associated instances can be retrieved only piece by piece, even distinguishing between  
3214 retrieval of association instances (e.g., via *GetReferencingInstance...*) and associated instances (e.g., via  
3215 *GetAssociatedInstance...*). Also, retrieving the associated instances associated by different associations  
3216 may involve the invocation of multiple class implementations in typical CIMOM/provider based  
3217 implementations, which could be optimized by having a single implementation of a more complex  
3218 operation like the one proposed here.

3219 An operation would be helpful that can retrieve the graph of associated instances including their  
3220 associations. Ideally, the operation would be able to traverse multiple association hops in one invocation.

3221 One possible definition of such operations could be:

3222 Direct retrieval: The *GetAssociatedGraphInstancesWithPath* operation traverses an association from an  
3223 instance on a source end to instances on all of its far ends and returns the associated instances and their  
3224 association instances, each together with their instance paths. This operation can be used to return one  
3225 set of instances that would have otherwise required at least twice as many operations (one set to get the  
3226 associations and another to get the related instances).

3227 Pulled retrieval: The *OpenAssociatedGraphInstancesWithPath* operation establishes and opens an  
3228 enumeration session for enumerating instances that are associated with the specified source instance,  
3229 and their association instances, including their instance paths. This operation can be used to return one  
3230 set of instances that would have otherwise required at least twice as many operations (one set to get the  
3231 associations and another to get the related instances).

## ANNEX B (informative)

### Change log

3232  
3233  
3234  
3235  
  
3236

Version	Date	Description
0.8.13	2007-07-11	Published as Work in Progress
1.0.0c	2008-08-26	Published as Preliminary Standard
1.0.0d	2009-11-02	<p>Published as Work in Progress, with the following changes:</p> <ul style="list-style-type: none"> <li>• Consolidated terminology with DSP0004 2.6 and DSP1001 1.1.</li> <li>• Simplified the definition of generic types by relating them to DSP0004 2.6.</li> <li>• Clarifications for error handling and for pre- and postconditions.</li> <li>• Added definition of ACID properties and defined ACID requirements on all operations.</li> <li>• CreateInstance: Fixed incorrect statement about initial value if a property defines no default value in its class declaration.</li> <li>• ModifyClass: Removed message WIPG0232.</li> <li>• OpenQueryInstances: Removed message WIPG0124.</li> <li>• OpenAssociatedInstances...: Replaced message WIPG0214 with WIPG0213.</li> <li>• OpenReferencingInstances...: Replaced message WIPG0214 with WIPG0213.</li> <li>• GetAssociatedInstances...: Added message WIPG0213.</li> <li>• GetReferencingInstances...: Added message WIPG0213.</li> <li>• Removed ExecQuery operation and QueryResult type.</li> <li>• Removed GetAssociatedGraphInstancesWithPath and OpenAssociatedGraphInstancesWithPath and added operations for retrieval of associated instance graphs into ANNEX A (Future operations).</li> <li>• Stated the messages to be used for precondition violations. This affects all operations.</li> <li>• Added sources of messages (infrastructure / class implementation). This affects all operations.</li> <li>• Added usage of message WIPG0249 as needed and adjusted the name of message WIPG0208, to accommodate the DSP8016 change that splits message WIPG0208 into WIPG0208 and WIPG0249. This affects most operations.</li> <li>• Removed informative annex about required updates to other DMTF specifications.</li> </ul>
1.0.0	2010-04-22	<p>Published as DMTF Standard, with the following changes:</p> <ul style="list-style-type: none"> <li>• Moved reference to DSP1001 into Bibliography</li> <li>• Changed terms: WBEM server, WBEM client, WBEM operation, WBEM protocol, WBEM listener, WBEM indication; Added references to document related terms in ISO guidelines.</li> <li>• Added "class implementation" as an additional source for error message WIPG0240 (WBEM service limits are exceeded) and WIPG0249 (Invalid input parameter value) on all instance related operations that use these messages</li> <li>• Generalized name of message WIPG0222 from "Query language feature not supported by WBEM service infrastructure" to "Query language feature not supported", following the corresponding change in DSP8016 1.0.1</li> <li>• Clarified that error message source (class specific vs. infrastructure) is a recommendation only</li> <li>• Changed DeleteDependents parameter of DeleteClass operation to be experimental</li> </ul>

Version	Date	Description
1.0.1	2012-08-30	<p>Published as DMTF Standard, with the following changes:</p> <ul style="list-style-type: none"><li>• 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).</li><li>• Clarified why the GetTopClassesWithPath operation does not have an IncludeInheritedElements parameter.</li></ul>

3237

## Bibliography

- 3238 DMTF DSP0200, *CIM Operations over HTTP 1.3*,  
3239 [http://www.dmtf.org/standards/published\\_documents/DSP0200\\_1.3.pdf](http://www.dmtf.org/standards/published_documents/DSP0200_1.3.pdf)
- 3240 DMTF DSP0201, *Representation of CIM in XML 2.3*,  
3241 [http://www.dmtf.org/standards/published\\_documents/DSP0201\\_2.3.pdf](http://www.dmtf.org/standards/published_documents/DSP0201_2.3.pdf)
- 3242 DMTF DSP0202, *CIM Query Language Specification 1.0*,  
3243 [http://www.dmtf.org/standards/published\\_documents/DSP0202\\_1.0.pdf](http://www.dmtf.org/standards/published_documents/DSP0202_1.0.pdf)
- 3244 DMTF DSP0203, *DTD for Representation of CIM in XML 2.3*,  
3245 [http://www.dmtf.org/standards/published\\_documents/DSP0203\\_2.3.dtd](http://www.dmtf.org/standards/published_documents/DSP0203_2.3.dtd)
- 3246 DMTF DSP0214, *Server Management Command Line Protocol Specification 1.0*,  
3247 [http://www.dmtf.org/standards/published\\_documents/DSP0214\\_1.0.pdf](http://www.dmtf.org/standards/published_documents/DSP0214_1.0.pdf)
- 3248 DMTF DSP0226, *Web Services for Management 1.0*,  
3249 [http://www.dmtf.org/standards/published\\_documents/DSP0226\\_1.0.pdf](http://www.dmtf.org/standards/published_documents/DSP0226_1.0.pdf)
- 3250 DMTF DSP0227, *WS-Management CIM Binding Specification 1.0*,  
3251 [http://www.dmtf.org/standards/published\\_documents/DSP0227\\_1.0.pdf](http://www.dmtf.org/standards/published_documents/DSP0227_1.0.pdf)
- 3252 DMTF DSP0230, *WS-CIM Mapping Specification 1.0*,  
3253 [http://www.dmtf.org/standards/published\\_documents/DSP0230\\_1.0.pdf](http://www.dmtf.org/standards/published_documents/DSP0230_1.0.pdf)
- 3254 DMTF DSP1001, *Management Profile Specification Usage Guide 1.1.0k (Work in Progress)*,  
3255 [http://www.dmtf.org/standards/published\\_documents/DSP1001\\_1.1.0.pdf](http://www.dmtf.org/standards/published_documents/DSP1001_1.1.0.pdf)
- 3256 JCP JSR-48, *Java Community Process JSR-48: WBEM Services Specification*, not yet published,  
3257 <http://jcp.org/en/jsr/detail?id=48>
- 3258 The Open Group CMPI, *Systems Management: Common Manageability Programming Interface 1.0*,  
3259 <http://www.opengroup.org/bookstore/catalog/c051.htm>