



1

2

3

4

**Document Number: DSP0227**

**Date: 2011-06-30**

**Version: 1.2.0**

5

## **WS-Management CIM Binding Specification**

6

**Document Type: Specification**

7

**Document Status: DMTF Standard**

8

**Document Language: en-US**

9

10 Copyright Notice

11 Copyright © 2006–2011 Distributed Management Task Force, Inc. (DMTF). All rights reserved.

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

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26 withdrawn or modified after publication, and shall be indemnified and held harmless by any party  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

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

32

# Contents

34	Foreword .....	7
35	Introduction .....	8
36	1 Scope .....	9
37	1.1 In-Scope.....	9
38	1.2 Out of Scope .....	9
39	1.3 Conformance .....	9
40	2 Normative References.....	9
41	3 Terms and Definitions.....	10
42	4 Symbols and Abbreviated Terms.....	11
43	5 Prefixes and XML Namespaces .....	12
44	6 WS-Management Default Addressing Model .....	13
45	6.1 Class-Specific ResourceURI.....	13
46	6.2 “All Classes” ResourceURI .....	15
47	6.3 Accounting for Different CIM Namespaces.....	15
48	7 Accessing Instances.....	16
49	7.1 Get .....	16
50	7.2 Put.....	16
51	7.3 Delete.....	17
52	7.4 Create .....	17
53	8 Filter Dialects.....	17
54	8.1 CQL.....	17
55	8.2 Association Queries .....	20
56	9 Enumeration .....	25
57	9.1 EnumerationMode.....	25
58	9.2 XmlFragment .....	26
59	9.3 Polymorphism .....	27
60	9.4 XPath Enumeration Using the Class-Specific ResourceURI.....	29
61	9.5 XPath Enumerate Using the “All Classes” ResourceURI .....	30
62	10 Subscriptions.....	30
63	10.1 Indication Filters.....	31
64	10.2 Subscribe Request.....	32
65	10.3 Subscription Response.....	36
66	10.4 Event Delivery.....	36
67	10.5 Subscription Reporting.....	37
68	10.6 Unsubscribe and Renew Requests .....	40
69	11 Extrinsic Methods .....	41
70	12 Exceptions.....	41
71	12.1 Fault Responses to Method Errors .....	41
72	12.2 Advertisement of Fault CIM_Error Inclusion.....	43
73	13 CIM Specific WS-Management Options.....	44
74	13.1 ShowExtensions Option.....	44
75	14 Instance Representation .....	45
76	15 Client Access to CIM Class Metadata.....	45
77	15.1 Applicability .....	45
78	15.2 Non-Separability of Metadata Access Functions.....	46
79	15.3 Overview of Metadata Operations .....	46
80	15.4 Targets of Metadata Operations .....	47
81	15.5 Class Metadata .....	47
82	15.6 Target Properties .....	47
83	15.7 Selectors .....	47

84 15.8 Options..... 48

85 15.9 EPR..... 50

86 15.10 Paths..... 51

87 15.11 Advertisement of CIM Class Metadata Path Types..... 51

88 15.12 Examples of Path EPR Containing URL..... 52

89 15.13 Example: Get CIM-XML Class Metadata for CIM\_ComputerSystem..... 54

90 15.14 Example: Enumerate EPRs for Class Metadata for CIM\_ComputerSystem and Classes

91 Derived from It ..... 55

92 15.15 Example: Enumerate WS-CIM Class Metadata for CIM\_ComputerSystem and Classes

93 Derived from It ..... 57

94 15.16 Example: Enumerate CIM-XML Class Metadata and EPRs for CIM\_ComputerSystem

95 and Classes Derived from It ..... 59

96 16 Fault Codes ..... 61

97 16.1 wsmb:CIMException ..... 62

98 16.2 wsmb:PolymorphismModeNotSupported ..... 62

99 17 Mapping for DSP0200 CIM Operations..... 62

100 17.1 Supported Operations..... 62

101 17.2 Unsupported Operations..... 72

102 18 Mapping of Error Messages to SOAP Fault Subcodes..... 72

103 19 XSD ..... 73

104 20 WSDL ..... 73

105 Bibliography ..... 76

106

107 **Tables**

108 Table 1 – Prefixes and XML Namespaces..... 12

109 Table 2 – CIM\_IndicationFilter Properties ..... 38

110 Table 3 – CIM\_ListenerDestinationWSManagement Required Properties ..... 38

111 Table 4 – CIM\_ListenerDestinationWSManagement Optional Properties..... 38

112 Table 5 – Required Properties for CIM\_IndicationSubscription and CIM\_FilterCollectionSubscription ..... 39

113 Table 6 – GenOps Operations and WS-Man Equivalents ..... 46

114 Table 7 – Targets Used in ResourceURI to Enumerate or Get Class Information ..... 47

115 Table 8 – Properties of a Class ResourceURI..... 47

116 Table 9 – Options That May Be Included in Operations Targeted at Metadata ..... 48

117 Table 10 – Examples of the Impact of Option Combinations on Operations Targeted at Metadata ..... 49

118 Table 11 – Elements of the EPR of an Operation Targeted at Metadata ..... 50

119 Table 12 – wsmb:CIMException ..... 62

120 Table 13 – wsmb:PolymorphismModeNotSupported ..... 62

121 Table 14 – GetInstance..... 63

122 Table 15 – GetInstance Arguments ..... 63

123 Table 16 – GetInstance Error Codes ..... 63

124 Table 17 – DeleteInstance ..... 64

125 Table 18 – DeleteInstance Arguments..... 64

126 Table 19 – DeleteInstance Error Codes..... 64

127 Table 20 – ModifyInstance..... 65

128 Table 21 – ModifyInstance Arguments ..... 65

129 Table 22 – ModifyInstance Error Codes ..... 65

130 Table 23 – CreateInstance..... 66

131 Table 24 – CreateInstance Arguments ..... 66

132 Table 25 – CreateInstance Error Codes ..... 66

133 Table 26 – EnumerateInstances ..... 66

134 Table 27 – EnumerateInstances Arguments..... 67

135 Table 28 – EnumerateInstances Error Codes..... 67

136 Table 29 – EnumerateInstanceNames ..... 67

137 Table 30 – EnumerateInstanceNames Arguments..... 68

138 Table 31 – EnumerateInstanceNames Error Codes..... 68

139 Table 32 – Associators..... 68

140 Table 33 – Associators Arguments ..... 68

141 Table 34 – Associators Error Codes ..... 69

142 Table 35 – AssociatorNames..... 69

143 Table 36 – AssociatorNames Arguments ..... 69

144 Table 37 – AssociatorNames Error Codes ..... 70

145 Table 38 – References..... 70

146 Table 39 – References Arguments ..... 70

147 Table 40 – References Error Codes ..... 71

148 Table 41 – ReferenceNames ..... 71

149 Table 42 – ReferenceNames Arguments..... 71

150 Table 43 – ReferenceNames Error Codes..... 71

151 Table 44 – CIM Error Messages with Corresponding Subcode Mappings ..... 72

152



154

## Foreword

155 The *WS-Management CIM Binding Specification* (DSP0227) was prepared by the DMTF WS-  
156 Management working group.

157 DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems  
158 management and interoperability.

### 159 **Acknowledgements**

160 The authors wish to acknowledge the following people.

#### 161 **Editors:**

- 162 • Nathan Burkhart, Microsoft
- 163 • Steve Hand, Symantec Corp.
- 164 • Richard Landau, Dell Inc.
- 165 • Hemal Shah, Broadcom Corporation

#### 166 **Contributors:**

- 167 • Josh Cohen, Microsoft Corporation (Chair)
- 168 • Doug Davis, IBM
- 169 • Jim Davis, WBEM Solutions
- 170 • David Hines, Intel
- 171 • Bryan Murray, Hewlett-Packard
- 172 • Brian Reistad, Microsoft Corporation
- 173 • Kirk Wilson, CA Inc.

174

175

## Introduction

176 This document describes the CIM binding for WS-Management. It describes how transformed CIM  
177 resources, as specified by the [WS-CIM Mapping Specification](#), are bound to WS-Management operations  
178 and WSDL definitions.

179



# 180           **WS-Management CIM Binding Specification**

## 181   **1   Scope**

182   This clause describes the scope of this specification, including some items that are specifically out of  
183   scope.

### 184   **1.1   In-Scope**

185   This specification describes how to use the Web Services for Management (WS-Management) protocol to  
186   communicate with resources modeled with CIM and exposed through the XML schema mapping described  
187   by WS-CIM.

### 188   **1.2   Out of Scope**

189   This specification does not describe how to expose the WBEM intrinsic methods that perform schema  
190   manipulation of CIM classes (for example, CreateClass) using the WS-Management protocol.

191   This specification does not describe how to generate the XML schema for a CIM class.

### 192   **1.3   Conformance**

193   This specification supplements the [WS-Management Specification](#). When this specification is supported,  
194   requests using a particular version of WS-Management are assumed to use the same version of this  
195   specification; both specifications will be updated concurrently. (The version of this specification cannot  
196   generally be directly determined from a SOAP message because most requests do not contain any  
197   elements from this specification or the XML namespace of this specification.)

198   An implementation is not conformant with this specification if it fails to satisfy one or more of the  
199   requirements defined in the conformance rules for each clause, as indicated by the following format:

200        **Rnnnn**: Rule text

## 201   **2   Normative References**

202   The following reference documents are indispensable for the application of this document. For dated  
203   references, only the edition cited applies. For undated references, the latest edition of the referenced  
204   document (including any amendments) applies.

205   DMTF DSP0004, *CIM Infrastructure Specification, 2.5*,  
206   [http://www.dmtf.org/standards/published\\_documents/DSP0004\\_2.5.pdf](http://www.dmtf.org/standards/published_documents/DSP0004_2.5.pdf)

207   DMTF DSP0200, *Specification for CIM Operations over HTTP, 1.3*,  
208   [http://www.dmtf.org/standards/published\\_documents/DSP0200\\_1.3.pdf](http://www.dmtf.org/standards/published_documents/DSP0200_1.3.pdf)

209   DMTF DSP0201, *Specification for the Representation of CIM in XML, 2.3*,  
210   [http://www.dmtf.org/standards/published\\_documents/DSP0201\\_2.3.pdf](http://www.dmtf.org/standards/published_documents/DSP0201_2.3.pdf)

211   DMTF DSP0203, *XML Document Type Definition, 2.3*,  
212   [http://www.dmtf.org/standards/published\\_documents/DSP0203\\_2.3.dtd](http://www.dmtf.org/standards/published_documents/DSP0203_2.3.dtd)

- 213 DMTF DSP0223, *Generic Operations Specification, 1.0*,  
214 [http://www.dmtf.org/standards/published\\_documents/DSP0223\\_1.0.pdf](http://www.dmtf.org/standards/published_documents/DSP0223_1.0.pdf)
- 215 DMTF DSP0226, *WS-Management Specification, 1.1*,  
216 [http://www.dmtf.org/standards/published\\_documents/DSP0226\\_1.1.pdf](http://www.dmtf.org/standards/published_documents/DSP0226_1.1.pdf)
- 217 DMTF DSP0230, *WS-CIM Mapping Specification, 1.0*,  
218 [http://www.dmtf.org/standards/published\\_documents/DSP0230\\_1.0.pdf](http://www.dmtf.org/standards/published_documents/DSP0230_1.0.pdf)
- 219 IETF RFC3986, *Uniform Resource Identifier (URI) Generic Syntax, January 2005*,  
220 <http://www.ietf.org/rfc/rfc3986.txt>
- 221 IETF RFC5646, *Tags for Identifying Languages, September 2009*,  
222 <http://tools.ietf.org/rfc/rfc5646.txt>
- 223 W3C, *Namespaces in XML, W3C Recommendations, 14 January 1999*,  
224 <http://www.w3.org/TR/1999/REC-xml-names-19990114>
- 225 W3C, *SOAP Version 1.2 Part 1: Messaging Framework (Second Edition) SOAP, 1.2, W3C*  
226 *Recommendation, 27 April 2007*,  
227 <http://www.w3.org/TR/soap12-part1/>
- 228 W3C, *Web Services Addressing 1.0 – Core, W3C Recommendation, May 2006*.  
229 <http://www.w3.org/TR/2006/REC-ws-addr-core-20060509/>
- 230 W3C, *Web Services Description Language (WSDL), 1.1, W3C Note, 15 March 2001*,  
231 <http://www.w3.org/TR/wsdl>
- 232 W3C, *XML Path Language (XPath) Version 1.0, W3C Recommendation, 16 November 1999*,  
233 <http://www.w3.org/TR/1999/REC-xpath-19991116>
- 234 W3C, *XML Schema Part 1: Structures Second Edition, W3C Recommendation, 28 October 2004*,  
235 <http://www.w3.org/TR/xmlschema-1/>

### 236 **3 Terms and Definitions**

237 The terms used in [DSP0226](#) and [DSP0230](#) also apply to this specification.

#### 238 **3.1**

##### 239 **can**

240 used for statements of possibility and capability, whether material, physical, or causal

#### 241 **3.2**

##### 242 **cannot**

243 used for statements of possibility and capability, whether material, physical or causal

#### 244 **3.3**

##### 245 **conditional**

246 indicates requirements to be followed strictly in order to conform to the document when the specified  
247 conditions are met

#### 248 **3.4**

##### 249 **mandatory**

250 indicates requirements to be followed strictly in order to conform to the document and from which no  
251 deviation is permitted

- 252 **3.5**  
253 **may**  
254 indicates a course of action permissible within the limits of the document
- 255 **3.6**  
256 **need not**  
257 indicates a course of action permissible within the limits of the document
- 258 **3.7**  
259 **optional**  
260 indicates a course of action permissible within the limits of the document
- 261 **3.8**  
262 **shall**  
263 indicates requirements to be followed strictly in order to conform to the document and from which no  
264 deviation is permitted
- 265 **3.9**  
266 **shall not**  
267 indicates requirements to be followed strictly in order to conform to the document and from which no  
268 deviation is permitted
- 269 **3.10**  
270 **should**  
271 indicates that among several possibilities, one is recommended as particularly suitable, without mentioning  
272 or excluding others, or that a certain course of action is preferred but not necessarily required
- 273 **3.11**  
274 **should not**  
275 indicates that a certain possibility or course of action is deprecated but not prohibited
- 276 **3.12**  
277 **unspecified**  
278 indicates that this profile does not define any constraints for the referenced CIM element or operation
- 279 **3.13**  
280 **base class**  
281 a class that is defined in a CIM schema and from which other classes are derived which may contain other  
282 properties or other CIM named elements  
283 These additional named elements are extensions to the base class.
- 284 **3.14**  
285 **addressing**  
286 the use of a web service specification to specify the address of a managed resource  
287 In this specification, two different versions of web service addressing may be used, depending on context  
288 and interoperability requirements. The general term "addressing" may be used to refer to Addressing  
289 defined in [WS-Management 1.1](#) Clause 5 or to [W3C Web Services Addressing 1.0](#).
- 290 **4 Symbols and Abbreviated Terms**
- 291 **4.1**  
292 **CQL**  
293 CIM Query Language

294	<b>4.2</b>
295	<b>EPR</b>
296	Endpoint Reference
297	<b>4.3</b>
298	<b>GED</b>
299	Global Element Declaration
300	<b>4.4</b>
301	<b>URI</b>
302	Uniform Resource Identifier
303	<b>4.5</b>
304	<b>WBEM</b>
305	Web-Based Enterprise Management
306	<b>4.6</b>
307	<b>WSDL</b>
308	Web Services Description Language
309	<b>4.7</b>
310	<b>XSD</b>
311	XML Schema Definition

## 312 **5 Prefixes and XML Namespaces**

313 Table 1 lists namespaces that are used in this specification. The choice of any namespace prefix is  
314 arbitrary and not semantically significant.

315 Note that two addressing prefixes are included. WS-Management 1.1 supports the use of two versions of  
316 addressing. In any particular protocol exchange, a single version of addressing is used. Examples in this  
317 specification generally specify one version or the other for clarity. In cases where the addressing version is  
318 not significant, examples use a non-version-specific "wsa:" prefix to indicate that either addressing version  
319 may be suitable in those cases, depending on the context of the message.

320

**Table 1 – Prefixes and XML Namespaces**

Prefix	XML Namespace	Reference
wsmb	<a href="http://schemas.dmtf.org/wbem/wsman/1/cimbinding.xsd">http://schemas.dmtf.org/wbem/wsman/1/cimbinding.xsd</a>	This specification
wsman	<a href="http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd">http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd</a>	WS-Management
cim	<a href="http://schemas.dmtf.org/wbem/wscim/1/common">http://schemas.dmtf.org/wbem/wscim/1/common</a>	WS-CIM
s	<a href="http://www.w3.org/2003/05/soap-envelope">http://www.w3.org/2003/05/soap-envelope</a>	SOAP 1.2
xs	<a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a>	XML Schema
wsdl	<a href="http://schemas.xmlsoap.org/wsdl">http://schemas.xmlsoap.org/wsdl</a>	WSDL 1.1
wsa04	<a href="http://schemas.xmlsoap.org/ws/2004/08/addressing">http://schemas.xmlsoap.org/ws/2004/08/addressing</a>	Addressing included in WS-Management 1.1 clause 5, "Addressing"
wsa10	<a href="http://www.w3.org/2005/08/addressing">http://www.w3.org/2005/08/addressing</a>	WS-Addressing 1.0

Prefix	XML Namespace	Reference
wsen	http://schemas.xmlsoap.org/ws/2004/09/enumeration	Enumeration included in WS-Management 1.1 clause 8, "Enumeration of Datasets"
wxf	http://schemas.xmlsoap.org/ws/2004/09/transfer	Resource access included in WS-Management 1.1 clause 7, "Resource Access"
wse	http://schemas.xmlsoap.org/ws/2004/08/eventing	Notifications included in WS-Management 1.1 clause 10, "Notifications (Eventing)"

## 321 6 WS-Management Default Addressing Model

322 WS-Management defines a default addressing model based on WS-Management 1.1 Addressing. This  
323 clause describes how CIM objects are addressed when they are accessed with the protocol.

324 WS-Management makes use of Addressing to identify and access resources. WS-Management defines a  
325 reference format using the EndpointReference element, making use of the ReferenceParameter field to  
326 contain specific elements (ResourceURI and SelectorSet) to aid in identifying the desired object or objects.

327 **R6-1:** Services that support the default addressing model defined by WS-Management are required  
328 to conform to this clause and its subclauses.

### 329 6.1 Class-Specific ResourceURI

330 For standard CIM classes, the ResourceURI is identical to the XML namespace URI of the schema for the  
331 class. This ResourceURI targets the named class and any derived classes depending on the role of  
332 polymorphism.

333 **R6.1-1:** Instances of a specific class shall be addressed using a ResourceURI that identifies a specific  
334 class.

335 **EXAMPLE:** The following ResourceURI is used to reference the CIM\_SoftwareElement class in version 2 of the CIM  
336 schema.

```
337 (01) http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_SoftwareElement
```

338 Note that the XML schema namespace for the instances never changes to reflect CIM namespace usage;  
339 only the ResourceURI changes. Class definitions are pure schema; they are independent of their scope or  
340 CIM namespace residence. See 6.3 for a description of classes that reside in explicit namespaces.

341 **R6.1-2:** It is recommended that vendor-defined classes use the same value for ResourceURI that is  
342 used for the XML namespace of the class. The vendor-defined XML namespace should include some  
343 form of version field in the namespace URI that can be changed when backward-incompatible changes  
344 are made to the XML schema.

345 Resources without keys are referenced by a class-specific ResourceURI within the SOAP binding, as  
346 follows:

```
347 (1) <s:Envelope ...>
348 (2) <s:Header>
349 (3) <wsa04:To> network address </wsa04:To>
350 (4) <wsman:ResourceURI> URI of the item </wsman:ResourceURI>
351 (5) </s:Header>
```

352 **R6.1-3:** If keys are required to discriminate among instances, the WS-Management SelectorSet SOAP  
 353 header shall be used, as follows:

```

354 (6) <s:Envelope ...>
355 (7) <s:Header>
356 (8) <wsa04:To> network address </wsa04:To>
357 (9) <wsman:ResourceURI> URI of the item </wsman:ResourceURI>
358 (10) <wsman:SelectorSet>
359 (11) <wsman:Selector Name="KeyName"> Key Value </wsman:Selector>
360 (12) </wsman:SelectorSet>
361 (13) ...
362 (14) </s:Header>

```

363 In this case, the key values required by CIM become individual Selector values. The name of the key is  
 364 repeated in the Name attribute, and the key value becomes the value of the Selector element. Note that all  
 365 CIM instances except indications have keys.

366 EXAMPLE: Example class definition:

```

367 (15) class CIM_SoftwareElement : CIM_LogicalElement
368 (16) {
369 (17) [key] string Name;
370 (18) [key] string Version;
371 (19) [key] uint16 SoftwareElementState;
372 (20) [key] string SoftwareElementID;
373 (21) [key] uint16 TargetOperatingSystem;
374 (22) string OtherTargetOS;
375 (23) string Manufacturer;
376 (24) string BuildNumber;
377 (25) string SerialNumber;
378 (26) string CodeSet;
379 (27) string IdentificationCode;
380 (28) string LanguageEdition;
381 (29) };

```

382 **R6.1-4:** The ResourceURI shall be the XML namespace for the class, and the zero or more Selectors  
 383 shall contain keys defined by this class. A service may process a request with a subset of the keys if  
 384 the subset uniquely identifies the instance. Clients are guaranteed correct behavior if they supply all  
 385 keys in the request. Clients might encounter different behavior at different resources if they do not  
 386 supply all keys.

387 EXAMPLE: The following example illustrates how to form an EPR using the class definition above:

```

388 (1) <s:Header xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
389 (2) <wsa04:To> network address </wsa04:To>
390 (3) <wsman:ResourceURI>
391 (4) http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_SoftwareElement
392 (5) </wsman:ResourceURI>
393 (6) <wsman:SelectorSet>
394 (7) <wsman:Selector Name="Name"> AcmeCAD </wsman:Selector>
395 (8) <wsman:Selector Name="Version"> 1.2 </wsman:Selector>
396 (9) <wsman:Selector Name="SoftwareElementState"> 1 </wsman:Selector>
397 (10) <wsman:Selector Name="SoftwareElementID"> 123F00 </wsman:Selector>
398 (11) <wsman:Selector Name="TargetOperatingSystem"> 12 </wsman:Selector>
399 (12) </wsman:SelectorSet>
400 (13) ...
401 (14) </s:Header>

```

402 **R6.1-5:** A service shall accept a properly-formed endpoint reference that specifies a class-specific  
403 ResourceURI and keys, if necessary, as defined in this clause.

404 CIM namespaces are not reflected in the ResourceURI structure, which is independent of where the class  
405 resides or is implemented.

## 406 6.2 “All Classes” ResourceURI

407 Because certain types of queries may cross class boundaries, the class-specific ResourceURI defined in  
408 6.1 is not always applicable.

409 **R6.2-1:** Services supporting cross-class queries shall accept an “all classes” ResourceURI.

410 This ResourceURI effectively targets the query processor in the CIM Server itself and can be used to return  
411 both CIM and vendor classes.

412 The “all classes” ResourceURI is of the same form as the class-specific ResourceURI in which the schema  
413 version and class name are replaced with the star character. The presence of the WS-CIM version in this  
414 ResourceURI allows the client to indicate which version of the [WS-CIM Mapping Specification](#) should be  
415 used in the translation of the CIM instances to XML.

416 For example, the following ResourceURI refers to all classes in the CIM namespace represented using  
417 version 1 of WS-CIM:

```
418 http://schemas.dmtf.org/wbem/wscim/1/*
```

419 When using the class-specific ResourceURI, the results of the enumeration may contain instances of the  
420 class identified in the ResourceURI or any derived class. However, the class name is typically repeated in  
421 both the ResourceURI and the filter expression.

422 The advantage to the “all classes” construct is that a single URI may be used for all resource queries and  
423 the class information appears in only one place: the filter expression. When the “all classes” construct is  
424 used in an Enumerate request, the results returned contain instances from a single CIM namespace, with  
425 one important exception: a query using an associationFilter filter dialect such as AssociatedInstances may  
426 return instances from more than one CIM namespace.

## 427 6.3 Accounting for Different CIM Namespaces

428 The following special Selector Name is defined to indicate the CIM namespace of the resource or  
429 resources for which the message is intended:

```
430 <wsman:Selector Name="__cimnamespace">xs:anyURI</wsman:Selector>
```

431 This selector is in addition to any other selectors for CIM keys and is unlikely to collide with others because  
432 most CIM keys do not start with two underscore (\_\_) characters.

433 The absence of this Selector Name in a message indicates that the intended resources are in the default  
434 CIM namespace for that service. This specification does not define what the default CIM namespace  
435 should be.

436 **R6.3-1:** A service offering more than one CIM namespace shall include the \_\_cimnamespace Selector  
437 Name in an EPR returned in a response message to identify the CIM namespace of an instance in the  
438 response.

439 **R6.3-2:** A service shall not fault if the \_\_cimnamespace Selector Name is absent and instead shall  
440 utilize the default CIM namespace.

441 **R6.3-3:** A service offering more than one CIM namespace should indicate in metadata which CIM  
442 namespace is the default. This specification does not define the location or format of such metadata.

443 **R6.3-4:** A service supporting more than one CIM namespace shall fault a request that specifies a  
444 namespace whose name is not one of the names of the CIM namespaces supported.

445 **R6.3-5:** If a service supports exactly one namespace, then

- 446 a. the service shall fault a request that includes a \_\_cimnamespace selector that does not  
447 match the name of the single namespace; and
- 448 b. the service should include the \_\_cimnamespace selector in an EPR returned in a response  
449 message to identify the CIM namespace of an instance in the response.

450 In all cases, **R6.3-2** applies: a request with no \_\_cimnamespace selector utilizes the default  
451 namespace. If a service supports only one namespace, then that namespace is the default.

## 452 7 Accessing Instances

453 When retrieving and updating an instance of a class, the WS-Management 1.1 Get, Put, and Delete  
454 operations are used. When creating an instance of a class, the Create operation is used. The fragment  
455 access SOAP header defined by WS-Management may be applied to these operations.

456 Class inheritance also affects how WS-Management 1.1 resource access operations are specified in WS-  
457 Management 1.1 clause 7, "Resource Access." In many cases vendors have derived a vendor-specific  
458 class from the CIM class that allows multiple vendors to implement the same class in the same CIM  
459 namespace even if they have not added any additional properties. For example, an implementation may  
460 choose to instantiate Vendor\_ComputerSystem, which is derived from CIM\_ComputerSystem. In many  
461 cases, a client must access instances of the derived class, but has only the name of the base class. To  
462 access an instance of such a derived class, or obtain an EPR for such an instance that can be used in WS-  
463 Management 1.1 resource access operations, a client generally will enumerate instances using the base  
464 class. The returned instances or EPRs can optionally contain the correct derived classname. See 9.3 for  
465 details.

466 The XML Schema representation of CIM instances permits the omission of non-key and non-required  
467 properties in their corresponding XML instance documents. The [WS-CIM Mapping Specification](#) (DSP0230)  
468 defines runtime rules for the Get, Delete, and Create operations.

469 **R7-1:** A service should return a wsa:ActionNotSupported fault if the "all classes" ResourceURI is  
470 used with any of the WS-Management 1.1 resource access operations, even if this ResourceURI is  
471 supported for enumerations or notifications.

### 472 7.1 Get

473 The following clause defines requirements and presents examples related to getting instances.

474 **R7.1-1:** A service supporting the Get operation and using the WS-Management Default Addressing  
475 Model shall support access using the class-specific ResourceURI that corresponds to the creation  
476 class and the selectors of the given instance.

477 **R7.1-2:** The response representation shall use the XML Schema identified by the class in the  
478 ResourceURI.

### 479 7.2 Put

480 The following clause defines requirements and presents examples related to putting or modifying instances.

481 **R7.2-1:** A service supporting the Put operation and using the WS-Management Default Addressing  
482 Model shall support access using the class-specific ResourceURI that corresponds to the creation  
483 class and the selectors of the given instance.



484 **R7.2-2:** A service supporting the Put operation shall accept instance representations that have omitted  
485 schema-optional elements. Any elements not included in the resource access operation shall be left  
486 unchanged. A service supporting fragment-level put operations shall also observe this behavior.

487 **R7.2-3:** The request and response representations shall use the XML Schema identified by the class in  
488 the ResourceURI.

### 489 7.3 Delete

490 The following clause defines requirements and presents examples related to deleting instances:

491 **R7.3-1:** A service supporting the Delete operation and using the WS-Management Default Addressing  
492 Model shall support access using the class-specific ResourceURI that corresponds to the creation  
493 class and the selectors of the given instance.

### 494 7.4 Create

495 The Create operation is different from the other WS-Management 1.1 resource access operations because  
496 it is sent to a resource factory rather than to a resource. For CIM, the class-specific ResourceURI is the  
497 factory resource that can be used to create instances of the class.

498 **R7.4-1:** A service supporting the Create operation and using the WS-Management Default Addressing  
499 Model shall support access using the class-specific ResourceURI corresponding to the creation class  
500 and, if warranted, the \_\_cimnamespace Selector Name.

501 However, the fragment-level Create operation operates on the resource itself, so it behaves in the same  
502 fashion as the Put operation:

503 **R7.4-2:** A service may support the fragment-level Create operation using the class-specific  
504 ResourceURI that corresponds to the creation class and the selectors of the given instance.

505 **R7.4-3:** A service supporting the Create operation shall accept instance representations that have  
506 omitted schema-optional properties and shall interpret such omissions as a request to create the object  
507 with the corresponding omitted properties absent from the instance. A service supporting the fragment-  
508 level Create operation shall also observe this behavior.

## 509 8 Filter Dialects

510 Both [WS-Management 1.1](#) enumeration and notifications define XPath Version 1.0 as the default filter  
511 language (called a "dialect" in those specifications), though other filter languages are accommodated. This  
512 specification defines two additional dialects for use with resources modeled using CIM. Services may  
513 support these and other query languages by accepting messages with appropriate dialect URIs.

514 The filter dialects defined in this clause are intended for use with WS-Management 1.1 Enumeration and  
515 WS-Management 1.1 notifications and not with Fragment-level WS-Management 1.1 resource access.

### 516 8.1 CQL

517 CQL is a SQL-based query language that includes the class name as part of the query. The dialect filter  
518 URI for this language is as follows:

519 `http://schemas.dmtf.org/wbem/cql/1/dsp0202.pdf`

520 **R8.1-1:** Services that accept CQL statements of the form "select \* from ..." shall return each instance  
521 representation using the GED defined for the object's class within the wsen:Items element.

522 **R8.1-2:** Services that accept CQL statements of the form "select a,b,c from ..." (a query with projection)

523 shall return each instance representation using the wsman:XmlFragment element. Within the  
 524 wsman:XmlFragment element, the service shall return property values named in the select statement  
 525 using either an element with the given label if the AS keyword is used or the property's GED defined in  
 526 the [WS-CIM Mapping Specification](#) if the select-list entry is a property (ignoring any chain or property-  
 527 scope). Expressions and literals without AS keywords are not valid CQL expressions.

528 Clients should use wsman:Filter, as opposed to wsen:Filter or wse:Filter, when using CQL statements of the  
 529 form "select a,b,c from ..." because these queries contain projections and are not Boolean predicates.

530 **R8.1-3:** Services supporting CQL statements of the form "select a,b,c from ..." may return results in any  
 531 order. To provide clients a mechanism to correlate results with the CQL expression, services should  
 532 include the attribute wsmb:Expression for all selected-entry elements, and shall include the attribute  
 533 wsmb:Expression for any selected-entry that would have a duplicate name with another selected-entry.  
 534 The value of the wsmb:Expression attribute on the element shall be the selected-entry in the select-list  
 535 from which the element resulted.

536 EXAMPLE 1: If the select-list of a CQL statement is "ID, Foo.Name, Bar::Host, A AS B, X \* Y AS Z", the query  
 537 returns the associated elements in the following fragment:

```
538 (1) <wsen:Items xmlns:ex='...'>
539 (2)   <wsman:XmlFragment>
540 (3)     <ex:ID>...</ex:ID>
541 (4)     <ex:Name>...</ex:Name>
542 (5)     <ex:Host>...</ex:Host>
543 (6)     <B>...</B>
544 (7)     <Z>...</Z>
545 (8)   </wsman:XmlFragment>
546 (9) </wsen:Items>
```

547 NOTE 1: The elements that result from the AS keyword do not have an XML namespace.

548 NOTE 2: Because the response elements are wrapped in the XmlFragment element, which is defined to turn off  
 549 validation for the entire content of the XmlFragment, it is permissible for the service not to include namespace prefixes  
 550 for the enclosed elements.

551 If a join were used with the same named property included from both classes, then the wsmb:Expression  
 552 would be used to differentiate between them.

553 EXAMPLE 2: Given a select-list of "CIM\_Foo.ID, CIM\_Foo.Name, CIM\_Bar.Name" the associated elements would  
 554 be as follows:

```
555 (1) <wsen:Items xmlns:bar='...' xmlns:foo='...'>
556 (2)   <wsman:XmlFragment>
557 (3)     <foo:ID>...</foo:ID>
558 (4)     <bar:Name wsmb:Expression='CIM_Bar.Name'> ...</bar:Name>
559 (5)     <foo:Name wsmb:Expression='CIM_Foo.Name'> ...</foo:Name>
560 (6)   </wsman:XmlFragment>
561 (7) </wsen:Items>
```

562 **R8.1-4:** If a service supports wsman:EnumerationMode=EnumerateObjectAndEPR for enumerating  
 563 instances and endpoint references, then it shall compose the instance representation of the results of  
 564 the CQL query (as specified in the previous two rules) with the EPR. The CQL query selects the  
 565 instances and properties of the instance to be returned but has no effect on the EPR that refers to  
 566 objects that match the where clause of the CQL query.

567 **R8.1-5:** If a service supports wsman:EnumerationMode=EnumerateEPR for enumerating endpoint  
 568 references, then it shall return the EPRs for instances that match the where clause of the CQL query  
 569 and ignore any properties specified in the select portion of the CQL query.

570 **R8.1-6:** If a service uses the WS-Management Default Addressing Model, then it should support this

571 filter dialect for Enumerate operations. If the CQL dialect is not supported by the addressed endpoint  
572 service, the service shall respond with a wsen:FilterDialectRequestedUnavailable fault.

573 **R8.1-7:** If a service uses the WS-Management Default Addressing Model and supports the CQL dialect  
574 for Enumerate operations it shall support addressing the CIM Server (through the "all classes"  
575 ResourceURI) and it should support addressing instances of a class (through the class-specific  
576 ResourceURI). If the CQL query references in the FROM clause more than one CIM class, then the  
577 Enumerate operation shall be addressed to the "all classes" ResourceURI. If the addressed endpoint  
578 and the query contradict each other (for example, the CIM classname in the class-specific  
579 ResourceURI does not match the CIM classname in the CQL FROM clause), the service shall respond  
580 with a wsen:CannotProcessFilter fault.

581 **R8.1-8:** If a service uses the WS-Management Default Addressing Model it should support this filter  
582 dialect for Subscribe operations. If the CQL dialect is not supported by the addressed endpoint service,  
583 the service shall respond with a wsen:FilterDialectRequestedUnavailable fault.

584 **R8.1-9:** If a service uses the WS-Management Default Addressing Model and supports the CQL dialect  
585 for Subscribe operations it shall support addressing the CIM Server (through the "all classes"  
586 ResourceURI) and it should support addressing instances of a class (through the class-specific  
587 ResourceURI). If the addressed endpoint and the query contradict each other (for example, the CIM  
588 classname in the class-specific ResourceURI does not match the CIM classname in the CQL FROM  
589 clause), the service shall respond with a wse:EventSourceUnableToProcess fault.

590 **R8.1-10:** Services that accept CQL queries should return instances of the most-derived class rather  
591 than a requested class, even though the query names a specific class.

592 **EXAMPLE 3:** The following request issues a CQL query in which the returned results include properties from the  
593 selected instances. This example uses the WS-Management Default Addressing Model but applies to  
594 any EPR model used by the service.

```
595 (1) <s:Envelope>
596 (2)   <s:Header>
597 (3)     <wsman:ResourceURI>
598 (4)       http://schemas.dmtf.org/wbem/wscim/1/*
599 (5)     </wsman:ResourceURI>
600 (6)   </s:Header>
601 (7)   <s:Body>
602 (8)     <wsen:Enumerate>
603 (9)       <wsman:Filter Dialect="http://schemas.dmtf.org/wbem/cql/1/dsp0202.pdf">
604 (10)        SELECT Name, PrimaryOwnerName
605 (11)        FROM CIM_ComputerSystem
606 (12)        WHERE EnabledState = 3
607 (13)      </wsman:Filter>
608 (14)    </wsen:Enumerate>
609 (15)  </s:Body>
610 (16) </s:Envelope>
```

611 The results include the two requested properties for instances that are "Disabled":

```
612 (1) <s:Body>
613 (2)   <wsen:PullResponse>
614 (3)     <wsen:EnumerationContext> ... </wsen:EnumerationContext>
615 (4)     <wsen:Items>
616 (5)       <wsman:XmlFragment>
617 (6)         <Name>system1</Name>
618 (7)         <PrimaryOwnerName>Joe</PrimaryOwnerName>
619 (8)       </wsman:XmlFragment>
```

```

620 (9)      <wsman:XmlFragment>
621 (10)     <Name>system2</Name>
622 (11)     <PrimaryOwnerName>Mary</PrimaryOwnerName>
623 (12)     </wsman:XmlFragment>
624 (13)     ... etc.
625 (14)     </wsen:Items>
626 (15)     </wsen:PullResponse>
627 (16)     </s:Body>

```

## 628 8.2 Association Queries

629 CIM uses associations to relate instances of different classes and defines intrinsic operations to find related  
630 classes. Association queries start with one instance that participates in the association (called the source  
631 object) and finds all related instances (called the result objects) linked through associations in which a  
632 reference to the source object appears as the value of a specific property (called the role) in the  
633 association. The query can be further constrained by limiting the roles that are used for the source or result  
634 objects as well as limiting the type of the association and result classes. Alternatively, it is possible to issue  
635 a query for instances of the associations themselves using a similar set of constraining parameters.

636 This specification defines the following dialect filter URI for association queries:

637 `http://schemas.dmtf.org/wbem/wsman/1/cimbinding/associationFilter`

638 The following rules apply only to services that support association queries:

639 **R8.2-1:** If a service uses the WS-Management Default Addressing Model it should support the  
640 association filter dialect for Enumerate operations that are addressed to the “all classes” ResourceURI.  
641 If such a service receives an Enumerate request addressed to a class-specific Resource URI  
642 specifying this filter dialect, the service shall respond with a `wsen:FilterDialectRequestedUnavailable`  
643 fault.

644 **R8.2-2:** If a service supports `wsman:EnumerationMode=EnumerateObjectAndEPR` for enumerating  
645 endpoint references, then it shall compose the instance representation of the results of the association  
646 query with the EPR as directed. The association query selects the instances and properties of the  
647 instance to be returned but has no effect on the presence or absence of the EPR.

648 **R8.2-3:** The service should return a `wse:FilteringRequestedUnavailable` fault in response to Subscribe  
649 requests using the association filter dialect.

650 **R8.2-4:** If the result set of a successful association query includes no instances, the service shall not  
651 return a fault.

### 652 8.2.1 Associated Instances

653 For queries that return associated instances, the Enumerate message has the following form:

```

654 (1) <wsen:Enumerate>
655 (2)   <wsman:Filter
656 (3)     Dialect="http://schemas.dmtf.org/wbem/wsman/1/cimbinding/associationFilter">
657 (4)     <wsmb:AssociatedInstances>
658 (5)       <wsmb:Object> xs:any </wsmb:Object>
659 (6)       <wsmb:AssociationClassName> xs:NCName </wsmb:AssociationClassName> ?
660 (7)       <wsmb:Role> xs:NCName </wsmb:Role> ?
661 (8)       <wsmb:ResultClassName> xs:NCName </wsmb:ResultClassName> ?
662 (9)       <wsmb:ResultRole> xs:NCName </wsmb:ResultRole> ?
663 (10)      <wsmb:IncludeResultProperty> xs:NCName </wsmb:IncludeResultProperty> *
664 (11)     </wsmb:AssociatedInstances>

```

665 (12) </wsman:Filter>

666 (13) </wsen:Enumerate>

667 The following definitions provide additional, normative constraints on the preceding outline:

- 668 • wsen:Enumerate/wsman:Filter/wsmmb:AssociatedInstances

669 The results include instances related to the source object through an association.

670 **R8.2.1-1:** The results of the enumeration shall be instances associated with the object through an  
671 association instance subject to the additional constraints listed in this clause.

- 672 • wsen:Enumerate/wsman:Filter/wsmmb:AssociatedInstances/wsmmb:Object

673 Identifies the source object for the association query and is required.

674 **R8.2.1-2:** The results shall be associated with the object identified by the endpoint reference in  
675 wsmmb:Object.

676 **R8.2.1-3:** If the EPR to which the Enumerate message is sent and the EPR of the source object  
677 reference two different CIM namespaces, the service may respond with a wsen:CannotProcessFilter  
678 fault.

679 **R8.2.1-4:** If the EPR of the source object does not reference exactly one valid CIM instance, the  
680 service shall respond with a wsen:CannotProcessFilter fault. Services should include a textual  
681 description of the problem.

- 682 • wsen:Enumerate/wsman:Filter/wsmmb:AssociatedInstances/wsmmb:AssociationClassName

683 Represents the name of a CIM association class. This element or parameter is optional.

684 **R8.2.1-5:** If the AssociationClassName is present, the results shall include only the instances related to  
685 the source object through associations that are instances of only the named class or derived classes. If  
686 the AssociationClassName is absent, results shall include instances that are related to the source  
687 object through associations of any type.

- 688 • wsen:Enumerate/wsman:Filter/wsmmb:AssociatedInstances/wsmmb:Role

689 Represents the name of a reference property of a CIM association class. This element or parameter is  
690 optional.

691 **R8.2.1-6:** If the Role name is present, the results shall include only instances related to the source  
692 object through an association in which the source object plays the specified role (that is, the name of  
693 the property in the association class that refers to the source object shall match the value of this  
694 parameter). If the Role name is absent, the results shall include instances associated to the source  
695 regardless of the role of the source object in the association.

- 696 • wsen:Enumerate/wsman:Filter/wsmmb:AssociatedInstances/wsmmb:ResultClassName

697 Represents the name of a CIM class. This element or parameters is optional.

698 **R8.2.1-7:** If the ResultClassName is present, the results shall include only objects that are instances of  
699 the named class or any of its derived classes. If the ResultClassName is absent, the results shall  
700 include all objects regardless of type.

- 701 • wsen:Enumerate/wsman:Filter/wsmmb:AssociatedInstances/wsmmb:ResultRole

702 Represents the name of a reference property of a CIM association class. This element or parameter is  
703 optional.

704 **R8.2.1-8:** If ResultRole name is present, the results shall only include instances related to the source  
705 object via an association in which the returned object plays the specified role. In other words, the name

706 of the property in the association class that refers to the returned object shall match the value of this  
707 parameter.

- 708 • wsen:Enumerate/wsman:Filter/wsemb:AssociatedInstances/wsemb:IncludeResultProperty

709 Represents the name of one or more properties of a CIM class. This element or parameter is optional.

710 **R8.2.1-9:** If the query does not include an IncludeResultProperty element, the service shall return each  
711 instance representation using the GED defined for the object's class within the wsen:Items element.

712 **R8.2.1-10:** If the query includes one or more IncludeResultProperty elements, the service shall  
713 return each instance representation using the wsman:XmlFragment element. Within the  
714 wsman:XmlFragment element, the service shall return property values using the property GEDs  
715 defined in the [WS-CIM Mapping Specification](#). If the query includes one or more  
716 IncludeResultProperty elements, the service shall not return any IncludeResultProperty elements not  
717 specified. The service shall ignore any IncludeResultProperty elements that describe properties not  
718 defined by the target class. If the service does not support fragment-level access, it shall return a  
719 wsman:UnsupportedFeature fault with the following detail code:

720 `http://schemas.dmtf.org/wbem/wsman/1/wsman/faultDetail/FragmentLevelAccess`

721 **R8.2.1-11:** A service may omit returned properties, even when explicitly requested, if and only if  
722 such properties have not been set (that is, the properties have a NULL value). The requestor is to  
723 interpret the absence of these properties as the properties having a NULL value.

724 **R8.2.1-12:** A service shall not return a fault if the association query contains a value for  
725 the AssociationClassName, Role, ResultClassName, or ResultRole method parameters that names a  
726 CIM element that is not defined in the target CIM namespace or relevant CIM class.

727 The association query uses these parameters to filter the results and not to define the results.

728 Clients should use wsman:Filter when using IncludeResultProperty elements because these queries  
729 contain projections and are not Boolean predicates.

730 EXAMPLE: The following request issues an association query in which the returned results include properties from  
731 the associated instances as well as the EPRs of the associated instances. This example uses the  
732 WS-Management Default Addressing Model but applies to any EPR model used by the service.

```
733 (1) <s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
734 (2)   xmlns:wsa04="http://schemas.xmlsoap.org/ws/2004/08/addressing"
735 (3)   xmlns:wsman="http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd"
736 (4)   xmlns:wsemb="http://schemas.dmtf.org/wbem/wsman/1/cimbinding.xsd"
737 (5)   xmlns:wsen="http://schemas.xmlsoap.org/ws/2004/09/enumeration">
738 (6)   <s:Header>
739 (7)     <wsman:ResourceURI>
740 (8)       http://schemas.dmtf.org/wbem/wscim/1/*
741 (9)     </wsman:ResourceURI>
742 (10)  </s:Header>
743 (11)  <s:Body>
744 (12)  <wsen:Enumerate>
745 (13)    <wsman:EnumerationMode>EnumerateObjectAndEPR</wsman:EnumerationMode>
746 (14)    <wsman:Filter
747 (15)      Dialect="http://schemas.dmtf.org/wsman/cimbinding/associationFilter">
748 (16)      <wsemb:AssociatedInstances>
749 (17)        <wsemb:Object>
750 (18)          <wsa04:Address> ... </wsa04:Address>
751 (19)          <wsa04:ReferenceParameters>
752 (20)          <wsman:ResourceURI>
```

```

753 (21) http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_PhysicalElement
754 (22) </wsman:ResourceURI>
755 (23) <wsman:SelectorSet>
756 (24) <wsman:Selector Name="Tag">81190b2</wsman:Selector>
757 (25) <wsman:Selector Name="CreationClassName">
758 (26) Vendor_PhysicalElement
759 (27) </wsman:Selector>
760 (28) </wsman:SelectorSet>
761 (29) </wsa04:ReferenceParameters>
762 (30) </wsmb:Object>
763 (31) <wsmb:AssociationClassName>
764 (32) CIM_SystemPackaging
765 (33) </wsmb:AssociationClassName>
766 (34) <wsmb:ResultClassName>CIM_System</wsmb:ResultClassName>
767 (35) <wsmb:IncludeResultProperty>Name</wsmb:IncludeResultProperty>
768 (36) <wsmb:IncludeResultProperty>
769 (37) PrimaryOwnerName
770 (38) </wsmb:IncludeResultProperty>
771 (39) </wsmb:AssociatedInstances>
772 (40) </wsman:Filter>
773 (41) </wsen:Enumerate>
774 (42) </s:Body>
775 (43) </s:Envelope>

```

The results include the two requested properties as well as the EPR of the associated instances:

```

776 (44) <s:Body>
777 (45) <wsen:PullResponse>
778 (46) <wsen:EnumerationContext> ... </wsen:EnumerationContext>
779 (47) <wsen:Items>
780 (48) <wsman:Item>
781 (49) <wsman:XmlFragment>
782 (50) <Name>system1</Name>
783 (51) <PrimaryOwnerName>Joe</PrimaryOwnerName>
784 (52) </wsman:XmlFragment>
785 (53) <wsa04:EndpointReference>
786 (54) <wsa04:Address> ... </wsa04:Address>
787 (55) <wsa04:ReferenceParameters>
788 (56) <wsman:ResourceURI>
789 (57) http://schemas.dmtf.org/cim/wscim/1/cim-schema/2/CIM_ComputerSystem
790 (58) </wsman:ResourceURI>
791 (59) ...
792 (60) </wsa04:ReferenceParameters>
793 (61) </wsa04:EndpointReference>
794 (62) </wsman:Item>
795 (63) <wsman:Item>
796 (64) <wsman:XmlFragment>
797 (65) <Name>system2</Name>
798 (66) <PrimaryOwnerName>Mary</PrimaryOwnerName>
799 (67) </wsman:XmlFragment>
800 (68) <wsa04:EndpointReference>
801 (69) <wsa04:Address> ... </wsa04:Address>
802 (70) <wsa04:ReferenceParameters>

```

```

804 (71) <wsman:ResourceURI>
805 (72)     http://schemas.vendor.com/.../Vendor_System
806 (73) </wsman:ResourceURI>
807 (74)     ...
808 (75)     </wsa04:ReferenceParameters>
809 (76) </wsa04:EndpointReference>
810 (77) </wsman:Item>
811 (78)     ...etc.
812 (79) </wsen:Items>
813 (80) </wsen:PullResponse>
814 (81) </s:Body>

```

## 815 8.2.2 Association Instances

816 For queries that return instances of the association class used in a relationship, the Enumerate message  
817 has the following form:

```

818 (1) <wsen:Enumerate>
819 (2)   <wsman:Filter
820 (3)     Dialect="http://schemas.dmtf.org/wbem/wsman/1/cimbinding/associationFilter">
821 (4)     <wsmb:AssociationInstances>
822 (5)       <wsmb:Object> xs:any </wsmb:Object>
823 (6)       <wsmb:ResultClassName> xs:NCName </wsmb:ResultClassName> ?
824 (7)       <wsmb:Role> xs:NCName </wsmb:Role> ?
825 (8)       <wsmb:IncludeResultProperty> xs:NCName </wsmb:IncludeResultProperty> *
826 (9)     </wsmb:AssociationInstances>
827 (10)  </wsman:Filter>
828 (11) </wsen:Enumerate>

```

829 The following definitions provide additional, normative constraints on the preceding outline:

- 830 • wsen:Enumerate/wsman:Filter/wsmb:AssociationInstances

831 The results include association instances related to the source object.

832 **R8.2.2-1:** The results of the enumeration shall be instances of an association class subject to the  
833 additional constraints listed in this clause.

- 834 • wsen:Enumerate/wsman:Filter/wsmb:AssociationInstances/wsmb:Object

835 Identifies the source object for the association query and is required.

836 **R8.2.2-2:** The results shall be instances of association classes for which one of the references is the  
837 object identified by this endpoint reference.

838 **R8.2.2-3:** If the EPR to which the Enumerate message is sent and the EPR of the source object  
839 represent two different CIM namespaces, the service may return a wsen:CannotProcessFilter fault.

840 **R8.2.2-4:** If the EPR of the source object does not reference exactly one valid CIM instance, the  
841 service shall respond with a wsen:CannotProcessFilter fault. Services should include a textual  
842 description of the problem.

- 843 • wsen:Enumerate/wsman:Filter/wsmb:AssociationInstances/wsmb:ResultClassName

844 Represents the name of a CIM association class. This element or parameter is optional.

845 **R8.2.2-5:** If the ResultClassName is present, the results shall contain only instances of the named  
846 class or a derived class.



- 847 • wsen:Enumerate/wsman:Filter/wsmb:AssociationInstances/wsmb:Role
- 848 Represents the name of a reference property of a CIM association class. This element or parameter is  
849 optional.
- 850 **R8.2.2-6:** If the Role element is present, the results shall include only instances of association classes  
851 that refer to the source object through a property whose name matches the value of this parameter.
- 852 • wsen:Enumerate/wsman:Filter/wsmb:AssociationInstances/wsmb:IncludeResultProperty
- 853 Represents the name of one or more properties of a CIM class. This element or parameter is optional.
- 854 **R8.2.2-7:** If the query does not include an IncludeResultProperty element, the service shall return each  
855 instance representation using the GED defined for the object's class within the wsen:Items element.
- 856 **R8.2.2-8:** If the query includes one or more IncludeResultProperty elements, the service shall return  
857 each instance representation using the wsman:XmlFragment element. Within the wsman:XmlFragment  
858 element, the service shall return property values using the property GEDs defined in the [WS-CIM  
859 Mapping Specification](#). If the query includes one or more IncludeResultProperty elements, the service  
860 shall not return any IncludeResultProperty elements not specified. The service shall ignore any  
861 IncludeResultProperty elements that describe properties not defined by the target class. If the service  
862 does not support fragment-level access, it shall return a wsman:UnsupportedFeature fault with the  
863 following detail code:
- 864 `http://schemas.dmtf.org/wbem/wsman/1/wsman/faultDetail/FragmentLevelAccess`
- 865 **R8.2.2-9:** A service may omit returned properties, even if explicitly requested, if and only if such  
866 properties have not been set (that is, the properties have a NULL value). The requestor is to interpret  
867 the absence of these properties as the properties having a value of NULL.
- 868 **R8.2.2-10:** A service shall not return a fault if the association query contains a value for the Role or  
869 ResultClassName method parameters that names a CIM element that is not defined in the target CIM  
870 namespace or relevant CIM class.
- 871 Clients should use wsman:Filter when using IncludeResultProperty elements as these queries contain  
872 projections and are not Boolean predicates.

## 873 9 Enumeration

874 [WS-Management 1.1](#) Enumeration is used as a basis for iteration through the members of a collection.  
875 When enumerating instances of classes, the WS-Management Enumerate operation is used.

### 876 9.1 EnumerationMode

877 Supporting wsman:EnumerationMode enables clients to use enumeration as a method to discover  
878 instances. Clients can incorporate one of the EnumerationMode values to obtain the endpoint reference to  
879 such instances.

880 **9.1-1:** To maximize interoperation, it is recommended that services that support enumeration also  
881 support wsman:EnumerationMode as defined in WS-Management.

882 **EXAMPLE 1:** The following example shows an unfiltered enumeration of a class. The class-specific ResourceURI is  
883 used when performing a simple unfiltered enumeration:

```
884 (1) ...
885 (2)   <s:Header>
886 (3)     <wsa04:Action>
887 (4)       http://schemas.xmlsoap.org/ws/2004/09/enumeration/Enumerate
```

```

888 (5)      </wsa04:Action>
889 (6)
890 (7)      <wsman:ResourceURI>
891 (8)          http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_ComputerSystem
892 (9)      </wsman:ResourceURI>
893 (10)     </s:Header>
894 (11)     <s:Body>
895 (12)         <wsen:Enumerate/>
896 (13)     </s:Body>

```

897 Enumerating this ResourceURI returns all instances of the named class and any derived classes:

```

898 (1) <CIM_ComputerSystem> <Name>Red-202</Name> ... </CIM_ComputerSystem>
899 (2) <CIM_ComputerSystem> <Name>Blue-03</Name> ... </CIM_ComputerSystem>
900 (3) <CIM_ComputerSystem> <Name>Blue-04</Name> </CIM_ComputerSystem>
901 (4) <Vendor_ComputerSystem> <Name>Green-1</Name> ... </Vendor_ComputerSystem>

```

902 Each XML instance retrieved by the preceding enumeration contains all the properties of the specific  
903 class. For example, the third XML instance is actually of type CIM\_UnitaryComputerSystem and might  
904 look as follows:

```

905 (1) <CIM_UnitaryComputerSystem
906 (2)     xmlns= "http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_UnitaryComputerSystem">
907 (3)
908 (4)     <Name> Blue-04 </Name>
909 (5)     <PowerManagementSupported> true </PowerManagementSupported>
910 (6)     <PrimaryOwnerName> Dave </PrimaryOwnerName>
911 (7)     ...
912 (8)
913 (9) </CIM_UnitaryComputerSystem>

```

## 914 9.2 XmlFragment

915 XPath allows fragments of the instance to be returned.

916 **9.2-1:** Some filter expressions allow fragments of the instance to be returned. When these ad-hoc  
917 queries are performed, the results should be wrapped using wsman:XmlFragment as per R7.7-1 of the  
918 [WS-Management Specification](#).

919 EXAMPLE 1: The following filter expression finds the name of all CIM\_ComputerSystems owned by Dave and  
920 returns just the Name element of the instance provided that the owner is "Dave":

```

921 XPath: ../CIM_ComputerSystem[PrimaryOwnerName="Dave"]/Name

```

922 The filter expression results in a PullResponse of the following form:

```

923 (1) <wsen:PullResponse>
924 (2)     <wsman:XmlFragment>
925 (3)         <Name> Red-202 </Name>
926 (4)     </wsman:XmlFragment>
927 (5)     <wsman:XmlFragment>
928 (6)         <Name> Blue-04 </Name>
929 (7)     </wsman:XmlFragment>
930 (8)     ...
931 (9) </wsen:PullResponse>

```

932 EXAMPLE 2: As a further refinement, just the value alone may be returned:

933 XPath: ../CIM\_ComputerSystem[PrimaryOwnerName="Dave"]/Name/text()

934 This modification of the filter expression results in a PullResponse of the following form:

```
935 (1) <wsen:PullResponse>
936 (2)   <wsman:XmlFragment> Red-202 </wsman:XmlFragment>
937 (3)   <wsman:XmlFragment> Blue-04 </wsman:XmlFragment>
938 (4)   ...
939 (5) </wsen:PullResponse>
```

### 940 9.3 Polymorphism

941 Many CIM implementations allow polymorphism.

942 A common way to extend CIM classes is to define derivatives of the CIM class. When a client requests  
943 objects of the type for CIM\_Process, it is possible to return instances that are actually of a derived type  
944 such as Vendor\_Process.

945 The result set may contain instances in accord with one of these three scenarios:

- 946 • Results should contain instances from the base class and all derived classes, and each  
947 instance should be represented in its actual type including any derived properties.
- 948 • Results should contain instances from the base class and all derived classes, but the XML  
949 document should be of the base class type and contain only elements corresponding to the  
950 properties of the base class.
- 951 • Results should contain only instances of the base class and no instances of derived classes.

952 The default behavior is to return all instances in their native representation.

953 **R9.3-1:** A service supporting enumeration shall include instances from the requested class and derived  
954 classes in the enumeration result unless otherwise directed by the client.

955 The client can request other behavior by adding the optional wsmb:PolymorphismMode element as a child  
956 element of the wsen:Enumerate element in the Enumeration request, as follows:

```
957 (
958   ...
959 (
960   <s:Body>
961   (
962     <wsen:Enumerate>
963     (
964       ...
965     (
966       <wsmb:PolymorphismMode> ... </wsmb:PolymorphismMode> ?
967     (
968       </wsen:Enumerate>
969     (
970 </s:Body>
```

971 **R9.3-2:** A service may optionally support the wsmb:PolymorphismMode modifier element with a value  
972 of ExcludeSubClassProperties. The ExcludeSubClassProperties PolymorphismMode shall return  
973 instances of the requested class and derived classes represented using the base class's GED and  
974 XSD type. Properties defined in the derived class are not returned.

975 **R9.3-3:** A service may optionally support the wsmb:PolymorphismMode modifier element with a value  
976 of None. The None Polymorphism mode shall return instances of the requested class only.

977 **R9.3-4:** A service may optionally support the wsmb:PolymorphismMode modifier element with a value

978 of IncludeSubClassProperties. The IncludeSubClassProperties shall return instances of the requested  
 979 class and derived classes using the actual class's GED and XSD type. This is the same as not  
 980 specifying the polymorphism mode.

981 **R9.3-5:** If the service does not support the requested polymorphism mode, it should return a  
 982 wsmb:PolymorphismModeNotSupported fault.

983 **R9.3-6:** The service should return a wsmb:PolymorphismModeNotSupported fault for requests using  
 984 the "all classes" ResourceURI if the PolymorphismMode element is present and does not have a value  
 985 of IncludeSubClassProperties.

986 **R9.3-7:** If both wsman:EnumerationMode and wsmb:PolymorphismMode are supported and  
 987 wsman:EnumerationMode is present in the request, the service shall always use the Resource URI of  
 988 the actual class in the returned EPR regardless of the value of wsmb:PolymorphismMode. This allows  
 989 the client to retrieve and update the actual instance.

990 **EXAMPLE 1:** The following example shows an unfiltered enumeration using just base class properties. Using the  
 991 PolymorphismMode element along with the class-specific ResourceURI yields the same results as the  
 992 example in 9.1, but the derived type is "cast away" or dropped.

```

993 (1) ...
994 (2)   <s:Header>
995 (3)     <wsa04:Action>
996 (4)       http://schemas.xmlsoap.org/ws/2004/09/enumeration/Enumerate
997 (5)     </wsa04:Action>
998 (6)
999 (7)     <wsman:ResourceURI>
1000 (8)       http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_ComputerSystem
1001 (9)     </wsman:ResourceURI>
1002 (10)  </s:Header>
1003 (11)  <s:Body>
1004 (12)    <wsen:Enumerate>
1005 (13)      <wsmb:PolymorphismMode> ExcludeSubClassProperties </wsmb:PolymorphismMode>
1006 (14)    </wsen:Enumerate>
1007 (15)  </s:Body>

```

1008 The same four instances are returned but "cast" as CIM\_ComputerSystem:

```

1009 (1) <CIM_ComputerSystem> <Name>Red-202</Name> ... </CIM_ComputerSystem>
1010 (2) <CIM_ComputerSystem> <Name>Blue-03</Name> ... </CIM_ComputerSystem>
1011 (3) <CIM_ComputerSystem> <Name>Blue-04</Name> ... </CIM_ComputerSystem>
1012 (4) <CIM_ComputerSystem> <Name>Green-1</Name> ... </CIM_ComputerSystem>

```

1013 Note that the third instance no longer contains the PowerManagementSupported property added by  
 1014 CIM\_UnitaryComputerSystem:

```

1015 (1) <CIM_ComputerSystem
1016 (2)   xmlns="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_ComputerSystem">
1017 (3)
1018 (4)   <Name> Blue-04 </Name>
1019 (5)   <PrimaryOwnerName> Dave </PrimaryOwnerName>
1020 (6)   ...
1021 (7)
1022 (8) </CIM_ComputerSystem>

```

1023 **R9.3-8:** If an Enumerate request specifies wsmb:PolymorphismMode=ExcludeSubClassProperties and  
 1024 wsman:EnumerationMode=EnumerateObjectAndEPR or EnumerateEPR, then the service shall return  
 1025 EPRs that reference instances of the most-derived classes of the requested class in the ResourceURI.

1026 The body of the request message appears as follows:

```
1027 (1) <wsen:Enumerate>
1028 (2)   <wsman:EnumerationMode> EnumerateObjectAndEPR </wsman:EnumerationMode>
1029 (3)   <wsmb:PolymorphismMode> ExcludeSubClassProperties </wsmb:PolymorphismMode>
1030 (4)   </wsen:Enumerate>
```

1031 The corresponding response message contains the following fragment. Note that the EPR for Blue-04 can be used to  
1032 access the property PrimaryOwnerName that is not present in the value returned.

```
1033 (1) <wsen:Items>
1034 (2)   <wsman:Item>
1035 (3)     <CIM_ComputerSystem> <Name>Red-202</Name> ... </CIM_ComputerSystem>
1036 (4)     <wsa04:EndpointReference>
1037 (5)       <wsa04:Address> ... </wsa04:Address>
1038 (6)       <wsa04:ReferenceParameters>
1039 (7)         <wsman:ResourceURI>
1040 (8)           http://schemas.dmtf.org/.../CIM_ComputerSystem
1041 (9)         </wsman:ResourceURI>
1042 (10)        <wsman:SelectorSet> ... </wsman:SelectorSet>
1043 (11)       </wsa04:ReferenceParameters>
1044 (12)     </wsa04:EndpointReference>
1045 (13)   </wsman:Item>
1046 (14)   <wsman:Item>
1047 (15)     <CIM_ComputerSystem> <Name>Blue-04</Name> ... </CIM_ComputerSystem>
1048 (16)     <wsa04:EndpointReference>
1049 (17)       <wsa04:Address> ... </wsa04:Address>
1050 (18)       <wsa04:ReferenceParameters>
1051 (19)         <wsman:ResourceURI>
1052 (20)           http://schemas.dmtf.org/.../CIM_UnitaryComputerSystem
1053 (21)         </wsman:ResourceURI>
1054 (22)        <wsman:SelectorSet> ... </wsman:SelectorSet>
1055 (23)       </wsa04:ReferenceParameters>
1056 (24)     </wsa04:EndpointReference>
1057 (25)   </wsman:Item>
1058 (26)   ...
1059 (27) </wsen:Items>
```

## 1060 9.4 XPath Enumeration Using the Class-Specific ResourceURI

1061 The ResourceURI contains the class name, as for unfiltered enumeration:

```
1062 (1) <wsman:ResourceURI>
1063 (2)   http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_ComputerSystem
1064 (3)   </wsman:ResourceURI>
```

1065 The XPath is anchored at an abstract array of CIM\_ComputerSystem XML nodes, which represent all  
1066 available instances:

```
1067 (1) <CIM_ComputerSystem> ... </CIM_ComputerSystem>
1068 (2) <CIM_ComputerSystem> ... </CIM_ComputerSystem>
1069 (3) <CIM_ComputerSystem> ... </CIM_ComputerSystem>
1070 (4) <CIM_ComputerSystem> ... </CIM_ComputerSystem>
```

1071 The XPath filter expression is evaluated against each possible instance of the specified class, and the  
 1072 instance is either selected as part of the result set or is discarded.  
 1073 PolymorphismMode=ExcludeSubClassProperties is used to ensure that all instances have the same type.

1074 The following XPath expressions all select every instance of CIM\_ComputerSystem and are identical:

```
1075 (1) XPath: .
1076 (2) XPath: ../CIM_ComputerSystem
```

1077 To filter, the [ ] filter expressions from XPath may be used. The following selects only instances that have a  
 1078 PrimaryOwnerName property set to "Dave":

```
1079 XPath: ../CIM_ComputerSystem[PrimaryOwnerName="Dave"]
```

1080 If PolymorphismMode=IncludeSubClassProperties were used, the following two XPath filters would have  
 1081 different results:

```
1082 (1) XPath: .[Owner="Dave"]
1083 (2) XPath: ../CIM_ComputerSystem[Owner="Dave"]
```

1084 The first XPath would match all instances regardless of type, while the second XPath would select only  
 1085 those instances whose actual type was CIM\_ComputerSystem.

## 1086 9.5 XPath Enumerate Using the "All Classes" ResourceURI

1087 As an alternative to a class-specific ResourceURI, the URI meaning "all classes" may be specified:

```
1088 http://schemas.dmtf.org/wbem/wscim/1/*
```

1089 This URI is a resource that refers to all instances of all classes. In this case, the abstract array of instances  
 1090 is mixed and includes elements of classes other than CIM\_ComputerSystem.

```
1091 (1) <CIM_ComputerSystem> ... </CIM_ComputerSystem>
1092 (2) <CIM_ComputerSystem> ... </CIM_ComputerSystem>
1093 (3) <CIM_SoftwareElement> ... </CIM_SoftwareElement>
1094 (4) <CIM_SoftwareElement> ... </CIM_SoftwareElement>
1095 (5) <CIM_LogicalDisk> ... <CIM_LogicalDisk>
1096 (6) <CIM_LogicalDisk> ... <CIM_LogicalDisk>
1097 (7) <CIM_LogicalDisk> ... <CIM_LogicalDisk>
1098 (8) ...etc.
```

1099 In the following example, the first query contains no class-specific information. Therefore, the query  
 1100 specifies "all instances of all classes". The second query refers to a specific class:

```
1101 (1) XPath: .
1102 (2) XPath: ../CIM_ComputerSystem
```

1103 Services do not typically support the first query if the "all classes" ResourceURI is used, but they may do  
 1104 so.

1105 NOTE: The XPath queries are identical to those provided in 9.4. The ResourceURI simply changes the implied pool  
 1106 of instances over which the query is executed.

## 1107 10 Subscriptions

1108 The WS-Management Subscribe operation (from [WS-Management 1.1](#) notifications) is used to subscribe to  
 1109 CIM indications. WS-Management 1.1 notifications uses the term "event" for the SOAP message sent to  
 1110 the receiver, while CIM uses the term "indication" for the observation of an event.

1111 The CIM Schema defines a set of special classes to support the delivery of indications to interested  
 1112 receivers. In the CIM Schema, indications are represented by the CIM\_Indication class or a subclass of  
 1113 CIM\_Indication. Subscriptions can express interest in a set of CIM\_Indications by providing a query  
 1114 expression or by referring to an already existing query. This clause outlines the relationship between the  
 1115 WS-Management 1.1 notifications messages and these CIM classes.

1116 A typical scenario for use of CIM indications would be a management client interested in receiving "sensor  
 1117 state change" indications from a device that it is managing. To receive these indications, the client would  
 1118 take the following steps:

- 1119 1) Construct or identify the indication filter.
- 1120 2) Create the WS-Management 1.1 notifications Subscribe request.
- 1121 3) Receive indications.

1122 A management service might need the ability to report on all subscriptions on a server.

1123 In the CIM Schema, subscriptions are represented by a trio of classes:

- 1124 • CIM\_IndicationFilter (or CIM\_FilterCollection) captures the query or filter identifying the  
 1125 subset of indications of interest.
- 1126 • CIM\_ListenerDestination captures information about where or how the indications are to be  
 1127 delivered.
- 1128 • CIM\_IndicationSubscription (or CIM\_FilterCollectionSubscription) associates an instance of  
 1129 CIM\_IndicationFilter (or CIM\_FilterCollection) with CIM\_ListenerDestination.

1130 These classes are used in different parts of the subscription life cycle, as indicated in the remainder of this  
 1131 clause.

1132 **R10-1:** A service that supports subscriptions shall do so using the WS-Management 1.1 notifications  
 1133 operations as defined in WS-Management. It is recommended that a service internally create the  
 1134 requisite CIM indication-related instances when the service accepts a subscription using the Subscribe  
 1135 message from a Web services client.

1136 **R10-2:** A service may deliver indications based on the creation of instances of the CIM indication-  
 1137 related classes in addition to supporting WS-Management 1.1 notifications.

1138 **R10-3:** A service that does not support the WS-Management Default Addressing Model is not required  
 1139 to conform to the rules for the ResourceURI described in the text and examples in the following  
 1140 subclauses (clause 10 and its subclauses). All examples about WS-Management 1.1 notifications filter  
 1141 dialects apply to services independent of their addressing model.

## 1142 10.1 Indication Filters

1143 When subscribing to indications, the same XPath and CQL filter usage is observed as for enumerations.  
 1144 However, association queries are not applicable to subscriptions.

1145 When CQL is used, the subscription filter includes the name of the class being selected for the  
 1146 subscription:

```
1147 select * from CIM_AlertIndication where MessageID="394"
```

1148 CQL statements with projections can also be used, in which case the selected properties of the indications  
 1149 are wrapped using wsman:XmlFragment as described in 8.1.

1150 The same filter can be expressed in XPath:

```
1151 ../CIM_AlertIndication[MessageID="394"]
```

1152 XPath filters can also be written without identifying the class. The same filter could be expressed using the  
 1153 following XPath filter if it were applied to instances of CIM\_AlertIndication:

```
1154      ./[MessageID="394"]
```

1155 These filter expressions can be formulated by the client, or they might already exist on the server (as an  
 1156 instance of CIM\_IndicationFilter).

## 1157 10.2 Subscribe Request

1158 The client constructs the subscribe request to express interest in a subset of the indications on the service.  
 1159 The client can filter the indications by specifying a filter directly in the subscribe request or by referring to an  
 1160 existing filter stored on the service.

### 1161 10.2.1 Subscribing Using a Filter

1162 When subscribing using a filter expression, the client can target the subscribe request to either the CIM  
 1163 Server or a specific indication class.

#### 1164 10.2.1.1 Subscribing to the CIM Server

1165 When subscribing to the CIM Server, a filter dialect such as CQL can be used. In this case, the query alone  
 1166 contains the necessary information as to which class is being filtered and the “all classes” ResourceURI  
 1167 can be used for addressing.

1168 **R10.2.1.1-1:** If a service supports client-supplied CQL expressions and the WS-Management Default  
 1169 Addressing Model, it should accept wse:Subscribe messages addressed to the “all-classes”  
 1170 ResourceURI.

1171 **EXAMPLE:** The following example shows a Subscribe message to set up a subscription for changes in sensor state.  
 1172 It is addressed to the “all classes” ResourceURI and uses a CQL filter to detect instance indications in  
 1173 which the CurrentState property has changed:

```
1174 (1) <s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
1175 (2)   xmlns:wsa04="http://schemas.xmlsoap.org/ws/2004/08/addressing"
1176 (3)   xmlns:wsman="http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd"
1177 (4)   xmlns:wse="http://schemas.xmlsoap.org/ws/2004/09/eventing">
1178 (5) <s:Header>
1179 (6)   <wsa04:Action>
1180 (7)     http://schemas.xmlsoap.org/ws/2004/08/eventing/Subscribe
1181 (8)   </wsa04:Action>
1182 (9)   <wsa04:To> http://127.0.0.1:9999/wsman </wsa04:To>
1183 (10)   <wsa04:MessageID> . . . </wsa04:MessageID>
1184 (11)   <wsa04:ReplyTo>
1185 (12)     http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
1186 (13)   </wsa04:ReplyTo>
1187 (14)   <wsman:ResourceURI>
1188 (15)     http://schemas.dmtf.org/wbem/wscim/1/*
1189 (16)   </wsman:ResourceURI>
1190 (17) </s:Header>
1191 (18) <s:Body>
1192 (19)   <wse:Subscribe>
1193 (20)     <wse:Delivery
1194 (21)       Mode="http://schemas.dmtf.org/wbem/wsman/1/wsman/PushWithAck">
1195 (22)     <wse:NotifyTo>
1196 (23)     <wsa04:Address> . . . </wsa04:Address>
```



```

1197 (24)      . . .
1198 (25)      </wse:NotifyTo>
1199 (26)      </wse:Delivery>
1200 (27)      <wsman:Filter dialect="http://schemas.dmtf.org/wbem/cql/1/dsp0202.pdf">
1201 (28)      <!-- whenever the state of any sensor changes -->
1202 (29)      SELECT *
1203 (30)      FROM CIM_InstIndication
1204 (31)      WHERE SourceInstance ISA CIM_Sensor
1205 (32)      AND PreviousInstance ISA CIM_Sensor
1206 (33)      AND PreviousInstance.CIM_Sensor::CurrentState <&t;&t;
1207 (34)      SourceInstance.CIM_Sensor::CurrentState
1208 (35)      </wsman:Filter>
1209 (36)      </wse:Subscribe>
1210 (37)      </s:Body>
1211 (38) </s:Envelope>

```

1212 When subscribing to the CIM Server, instances of all classes are implicitly addressed; therefore, separate  
 1213 polymorphism modes are not relevant.

1214 **R10.2.1.1-2:** A service supporting wse:Subscribe messages addressed to the “all classes”  
 1215 ResourceURI shall return a wsmb:PolymorphismModeNotSupported fault if the  
 1216 wsmb:PolymorphismMode modifier is present and does not equal IncludeSubClassProperties.

#### 1217 10.2.1.2 Subscribing to an Indication Class

1218 A subset of all indications can also be expressed by subscribing to an indication class. In this case, the  
 1219 EPR contains the necessary information as to which class is being filtered. An additional filter might or  
 1220 might not be present, but it would apply only to the instances of class indicated by the EPR.

1221 **R10.2.1.2-1:** If a service supports client filtering over a particular class of indications and the  
 1222 WS-Management Default Addressing Model, it should accept wse:Subscribe messages addressed to  
 1223 the class-specific ResourceURI for CIM\_Indication or a subclass of CIM\_Indication.

1224 **EXAMPLE:** The following example shows a Subscribe message to set up a subscription for changes in temperature  
 1225 sensors. It is addressed to the resource URI for the CIM\_AlertIndication class and uses XPath to select  
 1226 instances of the class in which one of the desired messages is present:  
 1227 Note that the NotifyTo EPR may specify either version of addressing, independent of the version used in  
 1228 the Subscribe message itself. See [DSP0226 1.1](#), clause 5.3, for clarification.

```

1229 (1) <s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
1230 (2)      xmlns:wsa04="http://schemas.xmlsoap.org/ws/2004/08/addressing"
1231 (3)      xmlns:wsman="http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd"
1232 (4)      xmlns:wse="http://schemas.xmlsoap.org/ws/2004/09/eventing" >
1233 (5) <s:Header>
1234 (6)   <wsa04:Action>
1235 (7)     http://schemas.xmlsoap.org/ws/2004/08/eventing/Subscribe
1236 (8)   </wsa04:Action>
1237 (9)   <wsa04:To> http://127.0.0.1:9999/wsman </wsa04:To>
1238 (10)  <wsa04:MessageID> . . . </wsa04:MessageID>
1239 (11)  <wsa04:ReplyTo>
1240 (12)    http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
1241 (13)  </wsa04:ReplyTo>
1242 (14)  <wsman:ResourceURI>
1243 (15)    http://schemas.dmtf.org/wbem/wscim/1/CIM_AlertIndication
1244 (16)  </wsman:ResourceURI>

```

```

1245 (17) </s:Header>
1246 (18) <s:Body>
1247 (19)   <wse:Subscribe>
1248 (20)     <wse:Delivery
1249 (21)       Mode="http://schemas.dmtf.org/wbem/wsman/1/wsman/PushWithAck">
1250 (22)     <wse:NotifyTo>
1251 (23)       <wsa:Address> . . . </wsa:Address>
1252 (24)       . . .
1253 (25)     </wse:NotifyTo>
1254 (26)   </wse:Delivery>
1255 (27)   <wsman:Filter
1256 (28)     xmlns:c="http://schemas.dmtf.org/wbem/wscim/1/CIM_AlertIndication">
1257 (29)     .[c:OwningEntity="DMTF" and (c:MessageID="394" or c:MessageID="396"
1258 (30)     or c:MessageID="398" or c:MessageID="400" or c:MessageID="413")]
1259 (31)   </wsman:Filter>
1260 (32) </wse:Subscribe>
1261 (33) </s:Body>
1262 (34) </s:Envelope>

```

1263 Additional filtering, such as XPath filters, on the instances of CIM\_AlertIndication that are identified by the  
 1264 EPR can be allowed. However, this practice is discouraged because using CQL expressions in this context  
 1265 creates the possibility for contradictions between the class identified by the EPR and the class identified in  
 1266 the CQL expression.

1267 **R10.2.1.2-2:** A service that supports a class-specific ResourceURI as a target of the wse:Subscribe  
 1268 message should return the wse:InvalidMessage fault if such messages specify a filter that includes  
 1269 class information as part of the filter expression.

1270 When the wse:Subscribe message is addressed to an indication class, the wsmb:PolymorphismMode  
 1271 element described in 9.3 can be used to control how polymorphism is handled for indications on event  
 1272 delivery. The wsmb:PolymorphismMode element becomes a child element of the Subscribe element.

1273 **R10.2.1.2-3:** A service supporting wse:Subscribe messages addressed to a CIM indication class  
 1274 through a class-specific ResourceURI shall provide indication instances from the requested class and  
 1275 its subclasses in event delivery unless otherwise directed by the client.

1276 **R10.2.1.2-4:** A service supporting wse:Subscribe messages addressed to a CIM indication class  
 1277 through a class-specific ResourceURI may support the use of the wsmb:PolymorphismMode modifier  
 1278 as a child of the wse:Subscribe element, with the resulting event instances typed according to rules  
 1279 **R9.3-2**, **R9.3-3**, and **R9.3-4**.

## 1280 10.2.2 Subscribing to an Existing Filter

1281 The service may have existing filters because of profile provisions implemented or filters previously created  
 1282 by a client. The client needs a way to express interest in one of these filters. These filters are represented  
 1283 by instances of either the CIM\_IndicationFilter or CIM\_FilterCollection classes; hereafter these instances  
 1284 are referred to as existing filters.

1285 **R10.2.2-1:** If a service supports filtering using an existing filter expression and the WS-Management  
 1286 Default Addressing Model, it should accept wse:Subscribe messages addressed to the class-specific  
 1287 ResourceURI for an instance of the existing filter class.

1288 **EXAMPLE:** The following example shows a Subscribe message to set up a subscription to an existing filter named by  
 1289 "example.org::temperatureSensors::stateChanges":

```

1290 (1) <s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
1291 (2)   xmlns:wsa04="http://schemas.xmlsoap.org/ws/2004/08/addressing"

```

```

1292 (3)   xmlns:wsman="http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd"
1293 (4)   xmlns:wse="http://schemas.xmlsoap.org/ws/2004/09/eventing" >
1294 (5)   <s:Header>
1295 (6)     <wsa04:Action>
1296 (7)       http://schemas.xmlsoap.org/ws/2004/08/eventing/Subscribe
1297 (8)     </wsa04:Action>
1298 (9)     <wsa04:To> http://127.0.0.1:9999/wsman </wsa04:To>
1299 (10)    <wsa04:MessageID> . . . </wsa04:MessageID>
1300 (11)    <wsa04:ReplyTo>
1301 (12)      http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
1302 (13)    </wsa04:ReplyTo>
1303 (14)    <wsman:ResourceURI>
1304 (15)      http://schemas.dmtf.org/wbem/wscim/1/CIM_IndicationFilter
1305 (16)    </wsman:ResourceURI>
1306 (17)    <wsman:SelectorSet>
1307 (18)      <wsman:Selector name="Name">
1308 (19)        example.org::temperatureSensors::stateChanges
1309 (20)      </wsman:Selector>
1310 (21)      <wsman:Selector name="SystemCreationClassName">
1311 (22)        CIM_ComputerSystem
1312 (23)      </wsman:Selector>
1313 (24)      <wsman:Selector name="__cimnamespace">interop</wsman:Selector>
1314 (25)    </wsman:SelectorSet>
1315 (26)  </s:Header>
1316 (27)  <s:Body>
1317 (28)    <wse:Subscribe>
1318 (29)      <wse:Delivery
1319 (30)        Mode="http://schemas.dmtf.org/wbem/wsman/1/wsman/PushWithAck">
1320 (31)      <wse:NotifyTo>
1321 (32)        <wsa:Address> . . . </wsa:Address>
1322 (33)        . . .
1323 (34)      </wse:NotifyTo>
1324 (35)    </wse:Delivery>
1325 (36)    <!-- wse:Filter and wsman:Filter not permitted in this case. -->
1326 (37)  </wse:Subscribe>
1327 (38) </s:Body>
1328 (39) </s:Envelope>

```

1329 **R10.2.2-2:** If a service supports filtering using an existing filter expression (as indicated by the EPR),  
1330 the service message shall return the wsman:InvalidParameter fault if the wse:Subscribe request  
1331 includes a filter expression (such as in the wse:Filter or wsman:Filter elements).

1332 **R10.2.2-3:** A service supporting Subscribe to an existing filter using the WS-Management Default  
1333 Addressing Model should support access using a class-specific ResourceURI corresponding to a filter  
1334 with selector values that identify the instance of the actual class of the desired filter. The referenced  
1335 base class shall be one for which CIM keys have been defined; otherwise, the service should respond  
1336 with a wsman:InvalidSelectors fault with the following detail code:

1337 <http://schemas.dmtf.org/wbem/wsman/1/wsman/faultDetail/UnexpectedSelectors>

1338 When subscribing to an existing filter, the classes of interest are indicated by the filter expression and  
1339 separate polymorphism modes are not relevant.

1340 **R10.2.2-4:** A service supporting wse:Subscribe messages addressed to an instance of  
 1341 CIM\_IndicationFilter or CIM\_FilterCollection through a class-specific ResourceURI shall return a  
 1342 wsmb:PolymorphismModeNotSupported fault if the wsmb:PolymorphismMode modifier is present and  
 1343 does not equal IncludeSubClassProperties.

1344 Subscribing to an instance of CIM\_IndicationFilter (or CIM\_FilterCollection) works regardless of whether or  
 1345 not the service created the filter or if a client constructed the instance prior to sending the Subscribe  
 1346 message. The client can construct instances of these filter classes using mechanisms such as WS-  
 1347 Management 1.1 resource access Create. In this case, the service is accepting a client-defined filter  
 1348 expression, so the service must also accept the same filter expression in a Subscribe message.

1349 **R10.2.2-5:** If a service supports creating an instance of CIM\_IndicationFilter (using WS-  
 1350 Management 1.1 resource access Create or another mechanism), the service shall also support a  
 1351 wse:Subscribe message in which the filter expression is specified in the wsman:Filter element in body  
 1352 of the Subscribe message.

### 1353 10.3 Subscription Response

1354 A successful SubscribeResponse message includes a SubscriptionManager element containing an EPR to  
 1355 be used to Unsubscribe from or Renew this subscription.

1356 **R10.3-1:** The SubscriptionManager EPR in a successful SubscribeResponse shall be unique, as seen  
 1357 by the Subscription Manager, to the subscription created by the Subscribe request.

1358 That is, the SubscriptionManager EPR returned by the service shall contain some elements that correlate,  
 1359 in the context of the Subscription Manager, one-to-one with the single subscription that was just created.

1360 **R10.3-2:** A service shall accept an Unsubscribe or Renew request whose EPR matches a  
 1361 SubscriptionManager EPR that was previously returned to a client, provided that the subscription is still  
 1362 active.

1363 That is, if a service accepts a subscription and returns a SubscriptionManager EPR to a client, the service  
 1364 shall accept that EPR as the target of an Unsubscribe or Renew message.

1365 Because both the client and the service depend on this EPR, the SubscriptionManager EPR shall be valid  
 1366 for the duration of the subscription.

### 1367 10.4 Event Delivery

1368 When instances of CIM\_Indication or a subclass are indicated by the notifications infrastructure, they are  
 1369 delivered as event SOAP messages according to the delivery mode in the wse:Subscribe request. The  
 1370 following rules describe the XML representation of the indication:

1371 **R10.4-1:** When delivering the event XML for an indication, the wsa:Action URI of the event should be  
 1372 set to the same value as the XML namespace for the actual class of the indication instance.

1373 **R10.4-2:** When delivering the event XML for an indication, the event body shall be the XML  
 1374 representation of the indication instance as per the [WS-CIM Mapping Specification](#), subject to any  
 1375 additional client requests such as projection or polymorphism.

1376 **EXAMPLE:** The following example shows an instance of CIM\_InstModification delivered as a single event using the  
 1377 Push delivery mode:

```
1378 (1) <s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
1379 (2)     xmlns:wsa04="http://schemas.xmlsoap.org/ws/2004/08/addressing"
1380 (3)     xmlns:wsman="http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd"
1381 (4)     xmlns:class=
1382 (5)         "http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_InstModification"
```

```

1383 (6)      xmlns:common="http://schemas.dmtf.org/wbem/wscim/1/common"
1384 (7)      xmlns:wse="http://schemas.xmlsoap.org/ws/2004/09/eventing">
1385 (8)      <s:Header>
1386 (9)          <wsa04:Action>
1387 (10)             http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_InstModification
1388 (11)          </wsa04:Action>
1389 (12)          <wsa04:To> . . . </wsa04:To>
1390 (13)          <wsa04:MessageID> . . . </wsa04:MessageID>
1391 (14)      </s:Header>
1392 (15)      <s:Body>
1393 (16)          <class:CIM_InstModification>
1394 (17)              <class:IndicationIdentifier>
1395 (18)                  CIM:12345678-abcd-0000-fedc-0123456789ab
1396 (19)              </class:IndicationIdentifier>
1397 (20)              <class:IndicationTime>
1398 (21)                  <common:dateTime>2007-04-01T11:22:33.123Z</common:dateTime>
1399 (22)              </class:IndicationTime>
1400 (23)              <class:PerceivedSeverity>5</class:PerceivedSeverity>
1401 (24)              <class:PreviousInstance> . . . </class:PreviousInstance>
1402 (25)              <class:SourceInstance> . . . </class:SourceInstance>
1403 (26)              <class:SourceInstanceHost>10.57.217.39</class:SourceInstanceHost>
1404 (27)              <class:SourceInstanceModelPath> . . . </class:SourceInstanceModelPath>
1405 (28)          </class:CIM_InstModification>
1406 (29)      </s:Body>
1407 (30) </s:Envelope>

```

## 1408 10.5 Subscription Reporting

1409 Subscription Reporting is the ability of an implementation to report on the existing filters, collections, and  
1410 subscriptions. Subscriptions can be created and deleted through the Subscribe and Unsubscribe  
1411 operations. Filters and subscriptions may also be created, modified, and deleted directly using other  
1412 protocol operations described in this specification. An implementation should instantiate instances that  
1413 reflect the results of the operations described in this specification.

1414 **R10.5-1:** It is recommended that a service create in its CIM service the requisite CIM indication-related  
1415 instances when the service accepts a subscription using the Subscribe message from a Web services  
1416 client. The CIM namespace in which these instances are created is beyond the scope of this  
1417 specification.

1418 The rules in the following clauses describe requirements for the content of the CIM indication-related  
1419 classes if such reporting is supported as recommended in the preceding rule.

1420 Every active subscription contains three components:

- 1421 • An instance of CIM\_IndicationFilter or CIM\_FilterCollection that describes the indications to  
1422 be delivered;
- 1423 • An instance of CIM\_ListenerDestinationWSManagement that describes the client-specified  
1424 endpoint for delivery of indications; and
- 1425 • An instance of CIM\_IndicationSubscription or CIM\_FilterCollectionSubscription that links the  
1426 filter and the destination, and describes additional characteristics of the subscription.

### 1427 10.5.1 CIM\_IndicationFilter

1428 The CIM\_IndicationFilter class captures the filter used in the subscription.

1429 **R10.5.1-1:** If a subscribe request contains a filter expression, a service shall create an instance of  
 1430 CIM\_IndicationFilter and set the properties as indicated in Table 2.

1431 **Table 2 – CIM\_IndicationFilter Properties**

Property Name	Value
Query	Filter expression from the Subscribe request, including XML if appropriate for the indicated QueryLanguage
QueryLanguage	Dialect URI from the Subscribe request For example, if a CQL expression were used in the Subscribe request the URI would be: <a href="http://schemas.dmtf.org/wbem/cql/1/dsp0202.pdf">http://schemas.dmtf.org/wbem/cql/1/dsp0202.pdf</a>

1432 When subscribing to an existing filter expression, the instance of CIM\_IndicationFilter already exists so a  
 1433 new instance is not created.

1434 **10.5.2 CIM\_ListenerDestinationWSManagement**

1435 The CIM\_ListenerDestinationWSManagement class captures the endpoint for event delivery.

1436 **R10.5.2-1:** A service shall ensure that, for each subscribed endpoint, an instance of  
 1437 CIM\_ListenerDestinationWSManagement exists and contains the properties as indicated in Table 3.

1438 **Table 3 – CIM\_ListenerDestinationWSManagement Required Properties**

Property Name	Value
Protocol	4 ("WS-Management")
Destination	The URL in the wsa:Address element of wse:NotifyTo If the delivery mode does not have a destination EPR (such as the Pull delivery mode), the WS-Management 1.1 Addressing or WS-Addressing anonymous URI should be used as a place holder. Using the anonymous URI indicates that the event sink will contact the event source; the anonymous URI is not to be confused with the ReplyTo EPR in that request.

1439 A WS-Management subscription contains a number of terms that extend the concept of a CIM subscription.  
 1440 Additional properties in CIM\_ListenerDestinationWSManagement capture these extensions. In most cases,  
 1441 the values of the new properties come from elements in the Subscribe request. In a few cases, the values  
 1442 are dictated by the WS-Management protocol.

1443 These properties are likely to be managed by users and client applications, and they might be of interest to  
 1444 users enumerating existing subscriptions. Some small footprint implementations of WS-Management  
 1445 services might not wish to expose all these properties.

1446 **R10.5.2-2:** If the subscribe request specifies any of the following options, the corresponding  
 1447 properties of the CIM\_ListenerDestinationWSManagement instance should be set according to the  
 1448 values shown in Table 4. These guidelines might be updated by newer versions of this class; the actual  
 1449 MOF definition takes precedence over the information in Table 4.

1450 **Table 4 – CIM\_ListenerDestinationWSManagement Optional Properties**

Property Name	Value
DestinationEndTo	Similar to Destination, but applies to the EndTo EPR, if present
Locale	<a href="#">RFC 5646</a> language code from the Subscribe request, if present
ContentEncoding	The value of the ContentEncoding element from the Subscribe request, if present

Property Name	Value
DeliveryMode	A ValueMap value that captures the Delivery/@Mode URI from the Subscribe request
Heartbeat	Interval in seconds at which point a heartbeat event will be sent if no other events have been sent
SendBookmarks	True if the SendBookmarks element was present in the Subscribe request
MaxTime	The time in seconds to build a batch when using a batching delivery mode
DeliveryAuth	The security profile URI being used by the event source when delivering events through a Push delivery mode
PolymorphismMode	A ValueMap value that captures the polymorphism choice if present in the Subscribe request

1451 In general, instances of ListenerDestinationWSManagement are not reusable because of the terms of the  
 1452 subscription and the rules regarding their deletion when a subscription ends. Whether instances are shared  
 1453 is beyond the scope of this specification.

1454 **10.5.3 CIM\_IndicationSubscription and CIM\_FilterCollectionSubscription**

1455 The CIM\_IndicationSubscription and CIM\_FilterCollectionSubscription classes capture associations  
 1456 between the indication filter or filter collection and the endpoint for event delivery. An instance of one of  
 1457 these classes represents the subscription created by the Subscribe request.

1458 **R10.5.3-1:** If a Subscribe request is addressed to an instance of CIM\_IndicationFilter, or results in  
 1459 the creation of an instance of CIM\_IndicationFilter, then a service shall create an instance of  
 1460 CIM\_IndicationSubscription and set the properties as indicated in Table 5 as part of a successful  
 1461 Subscribe operation.

1462 **Table 5 – Required Properties for CIM\_IndicationSubscription and CIM\_FilterCollectionSubscription**

Property Name	Value
SubscriptionDuration	The time at which the subscription expires as indicated in the Subscribe response
OnFatalErrorPolicy = "Remove"	Not applicable
RepeatNotificationPolicy = "None"	Not applicable
SubscriptionInfo	Unique value identifying the subscription

1463 **R10.5.3-2:** If a subscription request is addressed to an instance of CIM\_FilterCollection, then a  
 1464 service shall instead create an instance of CIM\_FilterCollectionSubscription with properties as  
 1465 indicated in Table 5.

1466 **R10.5.3-3:** If a service that supports Renew created an instance of CIM\_IndicationSubscription (or  
 1467 CIM\_FilterCollectionSubscription) when processing the Subscribe message, it shall update the  
 1468 SubscriptionDuration to reflect the new expiration time when processing the Renew message.

1469 WS-Management 1.1 notifications uses the subscription manager EPR in the SubscribeReponse message  
 1470 to identify the subscription. It defines the wse:Identifier element for use as a reference parameter in this  
 1471 EPR, but it is not required. For convenience, it is recommended that this element be used and match the  
 1472 SubscriptionInfo property.

1473 **R10.5.3-4:** A service should populate the SubscriptionInfo field with a URI to identify the  
 1474 subscription. If the wse:Identifier is being used as a reference parameter in the SubscriptionManager

1475 EPR, then the service should use the same value as the value of the wse:Identifier reference  
1476 parameter.

1477 Services can use the same URI format as outlined in 2.7 of the [WS-Management Specification](#) for  
1478 wsa:MessageID.

#### 1479 **10.5.4 Proxy Considerations**

1480 In some cases, the WS-Management service might be a proxy or adapter to an existing system. Such  
1481 implementations have the following two pieces of information to track:

- 1482 • the information about the subscription between the client and the WS-Management service
- 1483 • the information about the subscription between the WS-Management service and the CIM  
1484 Server

1485 The rules in this specification describe how to represent the information about the subscription between the  
1486 client and the WS-Management service. The representation of the information between the  
1487 WS-Management service and the CIM Server is beyond the scope of this specification.

1488 Implementations can choose to represent this “local” subscription using similar techniques, but the  
1489 information would differ in properties such as the CIM\_ListenerDestination.Destination that would be the  
1490 address of the WS-Management service for the local subscription. Implementations can choose to create  
1491 parallel subscriptions for each or do analysis to avoid sending the same indication multiple times on the  
1492 local channel.

#### 1493 **10.6 Unsubscribe and Renew Requests**

1494 A client may extend the duration of a subscription using a wse:Renew request, if the service supports such  
1495 requests.

1496 **R10.6-1:** If a service supports notifications but does not support renewing subscriptions, the service  
1497 may fault a wse:Renew request with the fault code wse:UnableToRenew. If a service supports  
1498 notifications, the service shall not fault a wse:Renew request with fault code wsa:ActionNotSupported

1499 Unsubscribe and Renew requests may be addressed to a service using the SubscriptionManager EPR that  
1500 was returned in the SubscribeResponse message.

1501 In lieu of using the SubscriptionManager EPR from the SubscribeResponse message, a client may  
1502 construct a new SubscriptionManager EPR of a particular form that is acceptable to the service. If the  
1503 ReferenceParameters of the EPR uniquely specify an existing instance of IndicationSubscription or  
1504 FilterCollectionSubscription, a service is required to accept the Unsubscribe or Renew request at the  
1505 normal protocol endpoint address, that is, the protocol endpoint where that subscription can be seen with  
1506 Enumerate or Get. The To address of the SubscriptionManager EPR is not necessarily valid over long  
1507 periods of time; the address may change because of dynamic addressing assigned to the protocol endpoint  
1508 or subscription manager service.

1509 **R10.6-2:** A service shall accept an Unsubscribe request or Renew request whose EPR specifies a valid  
1510 instance of IndicationSubscription or FilterCollectionSubscription. A service shall accept a request of  
1511 this form at the To address of the protocol endpoint at which the subscription can be accessed with  
1512 Enumerate or Get operations. A service may also accept a request of this form at the To address of the  
1513 SubscriptionManager EPR.

1514 If the EPR does not specify a valid and unique IndicationSubscription or FilterCollectionSubscription, then  
1515 the service shall fault the request. For instance, if a subscription has been terminated for any reason, then  
1516 a SubscriptionManager EPR or a constructed EPR specifying that subscription will not be valid.

1517 **R10.6-3:** A service shall delete at most one subscription as a result of an Unsubscribe request.



1518 The Unsubscribe request shall be sufficiently specific that it removes one subscription, or none in the case  
1519 of a fault for any reason.

1520 When a subscription is terminated, a service is required to clean up data structures that were created to  
1521 represent the subscription.

1522 When a subscriber is no longer interested in receiving indications from a subscription, it can cancel the  
1523 subscription using a wse:Unsubscribe request.

1524 **R10.6-4:** If a service created CIM indication-related instances as described in 10.5, then the service  
1525 shall delete those instances when the subscription is canceled for any reason.

1526 In all cases, the instance of CIM\_IndicationSubscription (or CIM\_FilterCollectionSubscription) is deleted  
1527 because this instance represents the actual subscription.

1528 Instances of the other members of the association might be reused between subscriptions. For example, if  
1529 a subscription were addressed to an existing filter (an instance of CIM\_IndicationFilter), then that instance  
1530 need not be deleted when the subscription is deleted. The exact ownership of these instances and a  
1531 method to determine when to delete them is beyond the scope of this specification.

## 1532 11 Extrinsic Methods

1533 Invoking an extrinsic method uses the action URIs and messages defined by the [WS-CIM Mapping](#)  
1534 [Specification](#) (clause 8.3, "CIM Methods to WSDL Mappings"). The request and response message  
1535 schemas for an extrinsic method are defined in the WS-CIM schema for the CIM class that defines the  
1536 method (and the request and response message schemas use the XML namespace for that class). The  
1537 wsa:Action URIs are derived from the XML namespace of the class and the method name as per the [WS-](#)  
1538 [CIM Mapping Specification](#). The endpoint reference is transformed into SOAP headers as defined by  
1539 [WS-Addressing](#) and [WS-Management 1.1](#), clause 5.1, in the same way as other WS-Management  
1540 elements.

1541 When using the WS-Management Default Addressing Model, the rules for ResourceURI and selector usage  
1542 are the same as those described in clause 7 of this specification.

## 1543 12 Exceptions

### 1544 12.1 Fault Responses to Method Errors

1545 For some CIM server implementations, invoking either an intrinsic or extrinsic method can result in the  
1546 production of one or more exceptions before the corresponding method completes on the CIM server. In  
1547 this case, the requested CIM operation may not be able to successfully complete and the service may not  
1548 be able to return the output for the operation. The service responds with a SOAP fault message containing  
1549 the exception instances according to the following rules:

1550 **R12.1-1:** If a service receives a WS-Management request message that translates into a CIM intrinsic  
1551 or extrinsic method, the execution of the method results in one or more exceptions, the requested CIM  
1552 operation does not complete, and the service is not able to return the output for the operation, the  
1553 service should respond with a SOAP fault.

1554 **R12.1-2:** A service responding to a WS-Management request that translated into a CIM intrinsic or  
1555 extrinsic method that did not complete and resulted in an exception should include each resultant  
1556 exception object as peers in the SOAP fault's Detail element. The XML representation of each  
1557 exception object shall conform to the mapping rules for CIM instances defined in the [WS-CIM Mapping](#)  
1558 [Specification](#).

1559 **R12.1-3:** A service responding to a WS-Management request that translated into a CIM intrinsic or

1560 extrinsic method that did not complete and resulted in an exception should use WS-Management fault  
 1561 subcodes that correspond to the nature of the exception that has occurred. If the exception does not  
 1562 correspond to any defined WS-Management fault subcode, the service should use the  
 1563 wsmb:CIMException subcode.

1564 For faults that return exception objects, the instances of the CIM\_Error in the env:Detail element has the  
 1565 following form:

```

1566 (1) <cimerr:CIM_Error
1567 (2)   xmlns:cimerr="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_Error"/>
1568 (3)   <cimerr:CIMStatusCode> . . . </cimerr:CIMStatusCode>
1569 (4)   <cimerr:Message> . . . </cimerr:Message>
1570 (5)   <cimerr:MessageArguments> . . . </cimerr:MessageArguments>
1571 (6)   <cimerr:MessageID> . . . </cimerr:MessageID>
1572 (7)   <cimerr:OwningEntity> . . . </cimerr:OwningEntity>
1573 (8)   <cimerr:PerceivedSeverity> . . . </cimerr:PerceivedSeverity>
1574 (9) . . . other properties as in WS-CIM . . .
1575 (10) </cimerr:CIM_Error>
  
```

1576 The following definitions provide additional, normative constraints on the preceding outline:

- 1577 • lines (1-2): cimerr:CIM\_Error

1578 **R12.1-4:** The instance shall be represented as the CIM\_Error class.

- 1579 • lines (3), (6), (8): cimerr:CIMStatusCode, cimerr:MessageID, cimerr:PerceivedSeverity

1580 These properties are required by the CIM schema.

1581 **R12.1-5:** The instance representation of CIM\_Error shall include all the properties required by the CIM  
 1582 Schema.

- 1583 • lines (4), (5), (7): cimerr:Message, cimerr:MessageArguments, cimerr:OwningEntity

1584 These properties are intended to be used by a client application to report an error in a user interface.  
 1585 In particular, MessageArguments combined with MessageID can be used to localize error messages  
 1586 for users.

1587 **R12.1-6:** It is recommended that the instance include values for these properties.

1588 **R12.1-7:** A service may include other properties of CIM\_Error in the instance representation.

1589 EXAMPLE: A fault response for an extrinsic method containing an invalid method parameter that results in a CIM  
 1590 exception would have the following structure:

```

1591 (1) <env:Fault>
1592 (2)   <env:Code>
1593 (3)     <env:Value>env:Sender</env:Value>
1594 (4)     <env:Subcode>
1595 (5)       <env:Value>wsman:InvalidParameter</env:Value>
1596 (6)     </env:Subcode>
1597 (7)   </env:Code>
1598 (8)   <env:Reason>
1599 (9)     <env:Text xml:lang="en">
1600 (10)       The invocation of CIM method RequestStateChange
1601 (11)       failed because the unknown parameter Spongebob
1602 (12)       has been supplied.
1603 (13)     </env:Text>
1604 (14)   </env:Reason>
  
```

```

1605 (15) <env:Detail>
1606 (16)   <wsman:FaultDetail>
1607 (17)     http://schemas.dmtf.org/wbem/wsman/1/wsman/faultDetail/InvalidName
1608 (18)   </wsman:FaultDetail>
1609 (19)   <cimerr:CIM_Error>
1610 (20)     <cimerr:CIMStatusCode>4</cimerr:CIMStatusCode>
1611 (21)     <cimerr:Message>RequestStateChange: Invalid input parameter "SpongeBob"
1612 (22)   </cimerr:Message>
1613 (23)     <cimerr:MessageArguments>SpongeBob</cimerr:MessageArguments>
1614 (24)     <cimerr:MessageID>ACME1234</cimerr:MessageID>
1615 (25)     <cimerr:OwningEntity>ACME:MyServer:ACME_PowerMgtSvc:1</cimerr:OwningEntity>
1616 (26)     <cimerr:PerceivedSeverity>7</cimerr:PerceivedSeverity>
1617 (27)     <cimerr:ProbableCause>130</cimerr:ProbableCause>
1618 (28)     <cimerr:ProbableCauseDescription>Unexpected
1619       input</cimerr:ProbableCauseDescription>
1620 (29)     . . . other properties as in WS-CIM . . .
1621 (30)   </cimerr:CIM_Error>
1622 (31) </env:Detail>
1623 </env:Fault>

```

1624 For further information on the mapping of CIM exceptions to WS-Management fault subcodes, see  
1625 clause 18.

1626 Services that support CIM\_Error may include classes derived from CIM\_Error, such as ACME\_Error, rather  
1627 than CIM\_Error itself. In order for a client to determine which XML element of the SOAP Fault Detail  
1628 represents CIM\_Error, this specification defines an XML attribute wsmb:IsCIM\_Error that has a type of  
1629 Boolean. The attribute shall only be used in the CIM\_Error or a derived class of CIM\_Error element.

1630 In practice, interoperability is best served when CIM\_Error service implementations include the attribute  
1631 with CIM\_Error or derived classes. No meaning may be inferred by the absence of the attribute.

1632 EXAMPLE: The IsCIM\_Error attribute may be used on a CIM\_Error element.

```
1633 <cimerr:CIM_Error wsmb:IsCIM_Error='true'> . . .
```

1634 **R12.1-8:** A service may include the IsCIM\_Error attribute with a value of true on a CIM\_Error (non-  
1635 derived class) element.

1636 **R12.1-9:** A service should include the IsCIM\_Error attribute with a value of true on a CIM\_Error derived  
1637 class element.

1638 **R12.1-10:** A Service should not include the IsCIM\_Error attribute on any element that does not  
1639 represent a CIM\_Error or derived class of CIM\_Error.

## 1640 12.2 Advertisement of Fault CIM\_Error Inclusion

1641 R12.1-2 indicates that a service should include the appropriate CIM\_Error elements in Faults that are  
1642 generated; however the service is not required to do so. There are situations in which clients will need to  
1643 know whether a service will include this information in advance of sending a request message. To enable  
1644 a client to detect this behavior, a service should advertise that it will send CIM\_Error elements in fault  
1645 messages by including a <Capability\_FaultIncludesCIMError> element within the WS-Management  
1646 IdentifyResponse message. The value of the <Capability\_FaultIncludesCIMError> is not meaningful and is  
1647 ignored

1648 EXAMPLE: The following fragment illustrates the inclusion of this additional element.

```
1649 (1) <wsmid:IdentifyResponse>
1650 (2)   <wsmid:ProtocolVersion>
```

```

1651 (3) http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd
1652 (4) </wsmid:ProtocolVersion>
1653 (5) . . .
1654 (6) <wsmb:Capability_FaultIncludesCIMError
1655 (7) xmlns:wsmb="http://schemas.dmtf.org/wbem/wsman/1/cimbinding.xsd"/>
1656 (8) . . .
1657 (9) </wsmid:IdentifyResponse>

```

1658 **R12.2-1:** A service that includes the <Capability\_FaultIncludesCIMError> element within an  
 1659 IdentifyResponse message shall include the appropriate CIM\_Error element or elements within the  
 1660 SOAP Faults it generates when it does not successfully process a CIM operation.

1661 NOTE: There may be reasons (e.g., security concerns) for a service to create but not transmit a SOAP Fault. The  
 1662 term "generate" is used to indicate that a SOAP Fault is created. However, the generation of a Fault is independent of  
 1663 whether it is transmitted, and transmission is determined by the implementation.

## 1664 13 CIM Specific WS-Management Options

1665 This specification relies on the WS-Management OptionSet extensibility mechanism for common scenarios.

### 1666 13.1 ShowExtensions Option

1667 Some of the optional CIM properties may be expensive to calculate; as a result, they are not included in  
 1668 casual queries for the resource representation. Also, in some CIM Server implementations, the CIM Server  
 1669 may define additional system properties that are stored along with the standard CIM properties of a given  
 1670 class and that are exposed using the open content model defined in the XML Schema specified in the [WS-  
 1671 CIM Mapping Specification](#).

1672 The use of ShowExtensions allows a client to indicate that the XML resource representation should contain  
 1673 the elements that are expensive to calculate and the extension elements, along with the rest of the  
 1674 resource properties. The ShowExtensions option may be applied to the WS-Management 1.1 resource  
 1675 access Get message, the WS-Management 1.1 Enumeration Enumerate message, and the WS-  
 1676 Management 1.1 notifications Subscribe message.

1677 When this option is applied to Enumerate, it communicates the desire for all resource representations  
 1678 returned by the enumeration sequence to include the extensions independent of whether they are returned  
 1679 in an EnumerateResponse or a PullResponse message.

1680 When this option is applied to a Subscribe message, it communicates the desire for all events matching  
 1681 that Subscribe message to be returned with the extensions.

1682 This specification does not define any meaning for the ShowExtensions option on other messages. If  
 1683 necessary, the client may place extra content in Put and Create messages using the extension mechanism  
 1684 defined in the [WS-CIM Mapping Specification](#).

1685 Because vendor extensions can be large or expensive to retrieve, a standard option has been defined to  
 1686 enable or disable the vendor extensions to be returned with the resource representation. The default is to  
 1687 disable the return of vendor extensions.

1688 To show all extensions, a client sets the Option value to ShowExtensions, as follows:

```

1689 (1) <wsman:OptionSet>
1690 (2) <wsman:Option name="ShowExtensions"/>
1691 (3) </wsman:OptionSet>

```

1692 To hide extensions, a client omits or sets the Option to FALSE or 0. Any other value or an empty element  
1693 implies that the extensions should be shown.

1694 **R13.1-1:** If a service receives a request with an OptionSet containing an Option named  
1695 ShowExtensions in which the OptionSet header has mustUnderstand="TRUE" and the Option element  
1696 has mustComply="TRUE" and the value of the Option element is FALSE or 0, the service shall return  
1697 the representation in minimal form or issue a fault.

1698 **R13.1-2:** If a service receives a request with an OptionSet containing an Option named  
1699 ShowExtensions in which the OptionSet header has mustUnderstand="TRUE" and the Option element  
1700 has mustComply="TRUE" and the value of the Option element is neither false nor 0, the service shall  
1701 return the representation with additional information including the cim:Key and cim:Version attributes as  
1702 per the [WS-CIM Mapping Specification](#) and any vendor-defined extensions or issue a fault.

1703 **R13.1-3:** In the absence of this option (or mustComply requirements), a service should return the  
1704 representation in minimal form or issue a fault.

1705 EXAMPLE: The following shows an example representation from a service that has implemented CIM schema  
1706 version 2.11.0 that includes extensions. Note that all the vendor-specific properties come after the class  
1707 properties.

```
1708 (1) <CIM_ComputerSystem
1709 (2)   xmlns="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_ComputerSystem "
1710 (3)   xmlns:cim="http://schemas.dmtf.org/wbem/wscim/1/common "
1711 (4)   xmlns:v="http://vendor.com/..."
1712 (5)   cim:Version="2.7.0">
1713 (6)
1714 (7)   <CreationClassName cim:Key="true"> ... </CreationClassName>
1715 (8)   <Name cim:Key="true"> Blue-04 </Name>
1716 (9)   <PrimaryOwnerName> Dave </PrimaryOwnerName>
1717 (10)  ...
1718 (11)  <v:PropertyCount>17</v:PropertyCount>
1719 (12) </CIM_ComputerSystem>
```

## 1720 14 Instance Representation

1721 Instances are represented according to the XML namespace defined by the [WS-CIM Mapping](#)  
1722 [Specification](#). This clause defines additional constraints on that representation.

1723 WS-CIM allows references to be represented using any version of Addressing. However, this specification  
1724 is associated with WS-Management, which requires that one of two specific addressing versions be used.

1725 **R14-1:** A service shall accept and return only instance representations in which XML elements  
1726 corresponding to CIM reference properties are represented as EPRs conformant to the requirements  
1727 defined in clause 6.

## 1728 15 Client Access to CIM Class Metadata

### 1729 15.1 Applicability

1730 Client applications using WS-Man may need access to the MOFs that define classes of management data.

1731 **R15.1-1:** A WS-Man service should provide class metadata using the mechanism described in this  
1732 clause.

## 1733 15.2 Non-Separability of Metadata Access Functions

1734 **R15.2-1:** If a service provides any class metadata operations described here, then all the normative  
1735 statements in clause 15 shall apply.

1736 For example, in order for a service to meet the requirements of this clause, the service must implement the  
1737 GetSubclassPaths option described in 15.3, and similarly for all other normative statements in this clause.

## 1738 15.3 Overview of Metadata Operations

1739 The WS-Management metadata operations are modeled after a subset of class operations in the *Generic*  
1740 *Operations Specification*, [DSP0223](#). The subset includes only operations to retrieve class metadata from a  
1741 service; a client cannot define new classes or modify classes using these operations.

1742 The metadata operations use existing WS-Management operations to retrieve class data from a service. A  
1743 client can use WS-Management Enumerate and Get operations to locate and retrieve metadata. These  
1744 operations are applied to special targets that retrieve class metadata rather than class instances. These  
1745 targets present special properties that are used as Selectors to identify the class.

1746 Class metadata can be retrieved in two forms:

- 1747 • The XML schema format (XSD) defined by [DSP0230](#) (WS-CIM); or
- 1748 • The XML format defined by [DSP0201](#) (CIM-XML).

1749 Additionally, services may support options that include or exclude specific pieces of metadata from the  
1750 result. In particular, because CIM classes are organized in a hierarchy, there are options to support  
1751 polymorphic retrieval of class and property metadata.

1752 The minimum requirements are very small to accommodate constrained implementations. For instance,  
1753 services may be able to respond only with the URL of the metadata requested and not with the full result  
1754 text. Such constrained implementations may support only a subset of the possible combinations of  
1755 options.

1756 The operations defined here are intended to parallel operations defined in the CIM *Generic Operations*  
1757 *Specification*, [DSP0223](#). Table 6 describes the WS-Management operations targeted for retrieving  
1758 metadata that are equivalent to certain Generic Operations.

1759 **Table 6 – GenOps Operations and WS-Man Equivalents**

Generic Ops Operation	WS-Man Operation Used	WS-Man Options Used
GetSubClassesWithPath	Enumerate	IncludePath, IncludePathEPR, IncludePathURL
GetSubClassPaths	Enumerate	IncludePath, IncludePathEPR, IncludePathURL, ExcludeClassSpecification
GetClass	Get	

1760 **R15.3-1:** A service shall implement the WS-Man equivalent of the GetSubclassPaths operation.

1761 Unless a service is very constrained with respect to memory and storage resources, it is strongly  
1762 recommended that the service implement all of these operations.

1763 **R15.3-2:** A service should implement the WS-Man equivalents of either the GetSubclassesWithPath  
1764 operation or the GetClass operation. A service may implement both operations.

1765 **15.4 Targets of Metadata Operations**

1766 **R15.4-1:** WS-Man operations that are targeted to retrieve metadata shall use the following targets to  
 1767 specify that the Enumerate or Get operations are intended to retrieve only class definition data and not  
 1768 class instances.

1769 These targets specify the syntax in which the class metadata is to be returned in the response message.  
 1770 An operation will always return the class metadata in the format requested unless the  
 1771 ExcludeClassSpecification option is specified.

1772 **Table 7 – Targets Used in ResourceURI to Enumerate or Get Class Information**

Target ResourceURI	Syntax of returned class data
http://schemas.dmtf.org/wbem/cim-xml/2/cim-schema/2/*	CIM-XML (XML document as defined in <a href="#">DSP0201</a> and <a href="#">DSP0203</a> )
http://schemas.dmtf.org/wbem/ws-cim/1/cim-schema/2/*	WS-CIM (XSD document as defined in <a href="#">DSP0230</a> )

1773 **R15.4-2:** A service shall provide class metadata in WS-CIM format, and should provide class metadata  
 1774 in CIM-XML format.

1775 **15.5 Class Metadata**

1776 The list of classes available at an endpoint may be a small subset of the CIM classes.

1777 **R15.5-1:** An endpoint shall contain the class metadata information of all classes for which instances  
 1778 might possibly appear in the endpoint.

1779 **R15.5-2:** A class named in a WS-Man operation targeted to retrieve metadata may be a class in the  
 1780 CIM schema or in an extension schema.

1781 **15.6 Target Properties**

1782 The targets in the table of ResourceURIs represent (synthetic) managed resources with two (synthetic)  
 1783 properties. These properties are used to select the metadata of specific classes.

1784 **Table 8 – Properties of a Class ResourceURI**

Property name	Property value
ClassName	The name of a class including schema name and classname within schema. Example: CIM_Sensor
ClassPath	The full WS-CIM URI for a class. Example: http://schemas.dmtf.org/wbem/ws-cim/1/cim-schema/2/CIM_Sensor

1785 **15.7 Selectors**

1786 **R15.7-1:** An operation targeted to retrieve metadata shall specify the name of the CIM class with either  
 1787 a ClassName property or a ClassPath property.

1788 **R15.7-2:** The wsman:SelectorSet element of an Enumerate or Get operation that is targeted to retrieve  
 1789 metadata shall include a Selector for exactly one of the properties ClassName or ClassPath. A service

1790 shall fault a request that includes Selectors for both ClassName and ClassPath.

1791 Note that the wsman:SelectorSet of an Enumerate operation that is targeted to retrieve metadata may be  
 1792 absent or empty; in this case the target is all classes.

1793 Classes are specific to CIM namespaces. A classname may appear in multiple CIM namespaces. The  
 1794 special Selector named "\_\_cimnamespace" is used to specify CIM namespaces in requests and responses.

1795 **R15.7-3:** The wsman:SelectorSet element may optionally include a Selector for the \_\_cimnamespace.

1796 The metadata of classes with the same name may be the same or different in different namespaces.

1797 **15.8 Options**

1798 Several options specifying the content of the returned metadata may be added to a class operation.  
 1799 These WS-Management options correspond to input parameters in the CIM [Generic Operations](#)  
 1800 [Specification](#). The names of the options shown in Table 9 are to be given as the value of the Name  
 1801 attribute of a wsman:Option element.

1802 **R15.8-1:** Zero or more of the options listed in Table 9 may be included in wsman:Option elements of a  
 1803 wsman:OptionSet element of a class operation, with the effect on the content of the response message  
 1804 as specified in the table. A single wsman:Option element shall include exactly one of these options by  
 1805 name.

1806 **Table 9 – Options That May Be Included in Operations Targeted at Metadata**

WS-Man Option	Used in Operations	Effect
IncludeClassOrigin	Enumerate, Get	If true, return in each element the name of the class in the hierarchy that defines the element. The syntax in which this information is returned depends on the syntax of the class definition.
IncludeQualifiers	Enumerate, Get	If true, return in each element the qualifiers declared in the MOF that defines the element. The syntax in which this information is returned depends on the syntax of the class definition.
IncludeSubclasses	Enumerate	If false, return the class and the first level of child classes derived directly from the class. If true, return class and all child classes derived from this class.
IncludeInheritedElements	Enumerate, Get	If false, return only elements defined in the class. If true, return all elements exposed in this class: that is, all elements defined in this class plus all inherited elements not overridden in this class.
IncludePath	Enumerate, Get	Return an element containing an EPR which can be used to retrieve the definition of the object. This is a synonym for the IncludePathEPR option.
IncludePathEPR	Enumerate, Get	Return an element containing an EPR which can be used to retrieve the definition of the object. For WS-Man operations, the path is an EPR to the class definition.
IncludePathURL	Enumerate, Get	Return an element containing a URL that can be dereferenced directly to retrieve the text of the desired metadata, using, for instance, a web browser. For WS-Man operations, this is a URL to the class definition.



WS-Man Option	Used in Operations	Effect
ExcludeClassSpecification	Enumerate	Do not return any elements describing the definition of the class, including metadata in either format, including Qualifiers, ClassOrigin elements or attributes, and InheritedElements. Paths will be returned if IncludePath is specified.  This option may be used to retrieve Paths only.

1807 **R15.8-2:** If an OptionSet block is marked with mustUnderstand="1", and an individual option is marked  
 1808 with MustComply="true", and the service cannot process that option, then the service shall fault the  
 1809 request as described in clause 6.4 of the [WS-Management Specification](#), "wsman:OptionSet."

1810 **R15.8-3:** At most one of the options specifying the form of returned path shall be specified in a single  
 1811 wsman:OptionSet; that is, only one of the set IncludePath, IncludePathEPR, and IncludePathURL shall  
 1812 be included in a single request.

1813 For example, it is possible that some metadata cannot be represented in a particular metadata syntax. If an  
 1814 option requests information to be included in the result that cannot be represented in the chosen syntax,  
 1815 then the service may fault the request.

1816 Note that in WS-Management all options have the value of "false" unless a value is explicitly stated as the  
 1817 value of the wsman:Option element. All the options defined here are Boolean. The value of any option is  
 1818 "false" unless "true" is explicitly stated as the value of the option. Consult the [WS-Management  
 1819 Specification \(DSP0226\)](#), clause 6.4, for clarification.

1820 Table 10 lists the impacts of some of the options. In the cases listed, an operation can choose to include or  
 1821 exclude in the response

- 1822 • Derived classes beyond the first level child classes;
- 1823 • Path EPRs or URLs; and
- 1824 • Class definition metadata.

1825 Not all combinations of options yield useful results for clients. For example, Enumerate with the  
 1826 combination of ExcludeClassSpecification="true" and IncludePath="false" will return no class metadata.  
 1827 Note that none of the options listed in Table 10 makes sense with Get operations.

1828 **Table 10 – Examples of the Impact of Option Combinations on Operations Targeted at Metadata**

WS-Man Operation	Include Subclasses Option	Path Option: IncludePath, IncludePathEPR, or IncludePathURL	Exclude Class Specification Option	Returned Class(es)	Returned Path EPR(s) or URL(s)
Enumerate	false	false	false	first level children	none
Enumerate	true	false	false	all children	none
Enumerate	true	true	false	all children	all children
Enumerate	false	true	false	first level children	first level children
Enumerate	false	true	true	none	first level children
Enumerate	true	true	true	none	all children

WS-Man Operation	Include Subclasses Option	Path Option: IncludePath, IncludePathEPR, or IncludePathURL	Exclude Class Specification Option	Returned Class(es)	Returned Path EPR(s) or URL(s)
Enumerate	true	false	true	none	none
Enumerate	false	false	true	none	none
Get	n/a	n/a	n/a	one class	none

1829

1830 Implementations may not be able to support all combinations of options. In particular, resource-constrained  
 1831 implementations that return only the EPR or URL of the metadata may not be able to support many  
 1832 combinations of options.

1833 • Example: If a service implementation returns URLs or EPRs that access static documents, the number  
 1834 of different documents for the different combinations of options may be limited. An implementation  
 1835 might support only a minimal format and an all-inclusive format. A "minimal" format could reflect WS-  
 1836 CIM mapping v1.0, IncludeClassOrigin=false, IncludeQualifiers=false, and  
 1837 IncludeInheritedElements=true. An "all-inclusive" format could include IncludeClassOrigin=true,  
 1838 IncludeQualifiers=true, and IncludeInheritedElements=true).

1839 • Example: A service implementation may include in the returned metadata more information than  
 1840 requested. For any options that are not declared "mustUnderstand" in the request, a service may  
 1841 ignore options that attempt to exclude some information from the result. In particular, this may be done  
 1842 to map requests to a limited number of available versions of the metadata.

1843 Rule R6.4-6 in the [WS-Management Specification](#) specifies the fault detail to be issued by a service that  
 1844 cannot support a required option.

1845 **15.9 EPR**

1846 **R15.9-1:** An EPR addressing a service that provides operations for retrieving metadata shall include  
 1847 the following elements.

1848 **Table 11 – Elements of the EPR of an Operation Targeted at Metadata**

Element	Value
To	URI of the WS-Man MAP endpoint, e.g., <ul style="list-style-type: none"> <li>• http://somedomain.tld:80/wsman</li> </ul>
Action	WS-Man action, one of <ul style="list-style-type: none"> <li>• http://schemas.xmlsoap.org/ws/2004/09/transfer/Get</li> <li>• http://schemas.xmlsoap.org/ws/2004/09/enumeration/Enumerate</li> </ul>
ReferenceParameters	ResourceURI element and SelectorSet element
ResourceURI	Target of the operation to retrieve metadata, one of <ul style="list-style-type: none"> <li>• http://schemas.dmtf.org/wbem/cim-xml/2/cim-schema/2/*</li> <li>• http://schemas.dmtf.org/wbem/ws-cim/1/cim-schema/2/*</li> </ul>

Element	Value
SelectorSet	<p>Selectors, exactly one specifying the class, either</p> <ul style="list-style-type: none"> <li>• &lt;wsman:Selector name="ClassName"&gt;CIM_Sensor&lt;/wsman:Selector&gt; or</li> <li>• &lt;wsman:Selector name="ClassPath"&gt;http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_Sensor&lt;/wsman:Selector&gt;</li> </ul> <p>and, optionally, one specifying the namespace</p> <ul style="list-style-type: none"> <li>• &lt;wsman:Selector name="__cimnamespace"&gt;interop&lt;/wsman:Selector&gt;</li> </ul>

1849 **15.10 Paths**

1850 Two types of paths to class data can be returned by a service: EPRs and URLs.

- 1851 • An EPR specifies a web service interface that a client can call as a web service to retrieve the  
1852 metadata using the WS-Management Get operation. That is, a client can use the EPR in a Get  
1853 operation to retrieve the particular class metadata.
- 1854 • A URL specifies a location on the web where the metadata can be retrieved using the operation  
1855 implied by the URL's scheme. Three schemes for URLs may be returned by services: HTTP, HTTPS,  
1856 and FTP. A client can dereference the URL using the operation implied by the scheme to retrieve the  
1857 particular class metadata.
- 1858 • A URL returned from a service is packaged inside an EPR, in the wsa:Address element of the  
1859 wsa:EndpointReference item. The URL is extracted from the EPR envelope in order to be used to  
1860 retrieve the metadata. Since the EPR designates a class and not an instance, no Selector elements  
1861 other than \_\_cimnamespace are needed, nor should be present, in such an EPR.

1862 An Enumerate operation can specify options to determine the types of paths returned: IncludePath,  
1863 IncludePathEPR and IncludePathURL.

1864 **R15.10-1:** A path returned by an Enumerate request targeting metadata and specifying an  
1865 IncludePath, IncludePathEPR, or IncludePathURL option shall be an EPR and shall follow the WS-  
1866 Management default addressing model.

1867 **R15.10-2:** A path returned by an Enumerate request targeting metadata and specifying the  
1868 IncludePathURL option shall contain a URL in the wsa:Address element of the returned EPR; and the  
1869 ResourceURI ReferenceParameter of the EPR shall contain a class-specific URI as described in 6.1.

1870 **R15.10-3:** A URL returned by an Enumerate request targeting metadata and specifying an  
1871 IncludePathURL option may employ any registered URI scheme. The URL may include path or query  
1872 string information or both to select the requested metadata.

1873 **R15.10-4:** If an Enumerate request targeting metadata does not specify an IncludePath,  
1874 IncludePathEPR, or IncludePathURL option, the service shall return no path information in the  
1875 response. To reduce the memory requirements for these functions in small footprint implementations, a  
1876 path EPR or URL returned by a service may specify the address of an endpoint other than the endpoint  
1877 to which the request operation was addressed.

1878 **15.11 Advertisement of CIM Class Metadata Path Types**

1879 To enable a client to determine whether a service supports the PathEPR or PathURL path types for CIM  
1880 metadata retrieval, a service should advertise that it will support these path types by including one or more  
1881 <Capability\_ClassMetadataPathType\_xxx> elements within the WS-Management IdentifyResponse  
1882 message. The value of each <Capability. . .> element will identify a supported path type. There are two

1883 defined elements: `Capability_ClassMetadataPathType_EPR` and  
 1884 `Capability_ClassMetadataPathType_URL`.

1885 EXAMPLE: The following fragment illustrates the inclusion of this additional element for a service that supports CIM  
 1886 metadata retrieval using the PathEPR path type.

```
1887 (10) <wsmid:IdentifyResponse>
1888 (11)   <wsmid:ProtocolVersion>
1889 (12)     http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd
1890 (13)   </wsmid:ProtocolVersion>
1891 (14)   . . .
1892 (15)   <wsmb:Capability_ClassMetadataPathType_EPR/>
1893 (16)   . . .
1894 (17) </wsmid:IdentifyResponse>
```

1895 EXAMPLE: The following fragment illustrates the inclusion of this additional element for a service that supports CIM  
 1896 metadata retrieval using either the PathEPR or the PathURL path type.

```
1897 (18) <wsmid:IdentifyResponse>
1898 (19)   <wsmid:ProtocolVersion>
1899 (20)     http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd
1900 (21)   </wsmid:ProtocolVersion>
1901 (22)   . . .
1902 (23)   <wsmb:Capability_ClassMetadataPathType_EPR/>
1903 (24)   <wsmb:Capability_ClassMetadataPathType_URL/>
1904 (25)   . . .
1905 (26) </wsmid:IdentifyResponse>
```

1906 **R15.11-1:** A service that includes the `<Capability_ClassMetadataPathType_EPR>` element within  
 1907 an `IdentifyResponse` message shall return EPRs in response to the `GetSubclassesWithPath` and  
 1908 `GetSubclassPaths` operations with either the `IncludePath` or `IncludePathEPR` option specified.

1909 **R15.11-2:** A service that includes the `<Capability_ClassMetadataPathType_URL>` element within  
 1910 an `IdentifyResponse` message shall return URLs in response to the `GetSubclassesWithPath` and  
 1911 `GetSubclassPaths` operations with the `IncludePathURL` option specified.

## 1912 15.12 Examples of Path EPR Containing URL

1913 The URL returned by a service to specify the metadata of a class may take many forms. Client  
 1914 applications should make no assumptions about the format or structure of the URL to be dereferenced.

1915 A service may return a URL that references, or appears to reference, a static file. The URL is embedded in  
 1916 an EPR that meets the requirements of the default addressing model.

```
1917 (1) <wsmen:Items>
1918 (2)   <wsman:Item>
1919 (3)     <wsa04:EndpointReference>
1920 (4)       <wsa04:Address>
1921 (5)         http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_ComputerSystem.xsd
1922 (6)       </wsa04:Address>
1923 (7)       <wsa04:ReferenceParameters>
1924 (8)         <wsman:ResourceURI>
1925 (9)           http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_ComputerSystem
1926 (10)        </wsman:ResourceURI>
```

```

1927 (11)         <wsman:SelectorSet>
1928 (12)             <wsman:Selector name="__cimnamespace">root/cimv2</wsman:Selector>
1929 (13)         </wsman:SelectorSet>
1930 (14)         </wsa04:ReferenceParameters>
1931 (15)         </wsa04:EndpointReference>
1932 (16)     </wsman:Item>
1933 (17) </wsman:Items>

```

1934 Notes on this example:

- 1935 • (lines 4-10) The path of the URL specifies an XSD file using its full filename. The class ResourceURI  
1936 does not include the .xsd extension of the filename.
- 1937 • (line 9) The EPR includes the ResourceURI of the class. It is not required that the URL be of a form  
1938 where the classname is easily parsed. Therefore, to enable the client to distinguish the several results  
1939 of an Enumerate operation, each EPR must specify the class that the metadata will represent.
- 1940 • (lines 11-13) The EPR includes the CIM namespace that may contain instances of the class that the  
1941 metadata will represent. The combination of CIM namespace and CIM classname uniquely specify the  
1942 metadata of the class.

1943 A returned URL may specify FTP as the transport mechanism. If no user information and password are  
1944 included in the URL, anonymous FTP is assumed.

```

1945 (1) <wsa04:Address>
1946 (2) ftp://schemas.bitsrus.com/wbem/ws-cim/1/cim-schema/2/CIM_Sensor.xsd
1947 (3) </wsa04:Address>

```

1948 A library of static metadata files may be organized into directories that specify the set of options included in  
1949 the dereferenced metadata.

```

1950 (1) <wsa04:Address>
1951 (2) ftp://schemas.bitsrus.com/wbem/ws-cim/1/cim-schema/2/withqualifiers/CIM_Sensor.xsd
1952 (3) </wsa04:Address>

```

1953 Metadata may be requested in CIM-XML format as well as WS-CIM XSD format.

```

1954 (1) <wsa04:Address>
1955 (2) ftp://schemas.bitsrus.com/wbem/cim-xml/cim-schema/2/complete/CIM_Sensor.xml
1956 (3) </wsa04:Address>

```

1957 Metadata may be served by web applications that use the query string of the URL to specify the class.

```

1958 (1) <wsa04:Address>
1959 (2) http://schemas.bitsrus.com/cimv2/xsd_with_all_qualifiers.php?classname=CIM_Sensor
1960 (3) </wsa04:Address>
1961 (4)
1962 (5) <wsa04:Address>
1963 (6) http://schemas.bitsrus.com/cimv2/xsd_minimal.php?classname=CIM_Sensor
1964 (7) </wsa04:Address>

```

1965 A URL may specify some or all options in the query string. This method probably requires the least effort  
1966 for small footprint services.

```

1967 (32) <wsa04:Address>
1968 (33) http://schemas.bitsrus.com/cimv2/select_xsd.php?classname=CIM_Sensor&IncludeQualifiers=true&I

```

```

1969     includeInheritedElements=true&IncludeClassOrigin=true
1970 (34) </wsa04:Address>

```

### 1971 **15.13 Example: Get CIM-XML Class Metadata for CIM\_ComputerSystem**

1972 The following XML fragment illustrates the use of the ResourceURI, Selectors, and Options to specify an  
 1973 operation targeted to retrieve metadata.

```

1974 (1) <!-- Example fragment of XML for a get class operation. -->
1975 (2) <env:Envelope>
1976 (3) <env:Header>
1977 (4) <wsa04:To>http://somedomain.tld:80/wsman</wsa04:To>
1978 (5) <wsa04:Action>http://schemas.xmlsoap.org/ws/2004/09/transfer/Get</wsa04:Action>
1979 (6) <wsman:ResourceURI>
1980 (7) http://schemas.dmtf.org/wbem/cim-xml/2/cim-schema/2/*
1981 (8) </wsman:ResourceURI>
1982 (9) <wsman:SelectorSet>
1983 (10) <wsman:Selector Name="ClassName">CIM_ComputerSystem</wsman:Selector>
1984 (11) <wsman:Selector Name="__cimnamespace">root/interop</wsman:Selector>
1985 (12) </wsman:SelectorSet>
1986 (13) <wsman:OptionSet mustUnderstand="true">
1987 (14) <wsman:Option Name="IncludeInheritedElements"
1988 MustComply="true">true</wsman:Option>
1989 (15) <wsman:Option Name="IncludeQualifiers" MustComply="true">true</wsman:Option>
1990 (16) </wsman:OptionSet>
1991 (17) </env:Header>
1992 (18) <env:Body>
1993 (19) </env:Body>
1994 (20) </env:Envelope>

```

1995 The request includes the following elements.

- 1996 • (line 5) The Action specifies to Get the class metadata.
- 1997 • (line 6) The ResourceURI specifies that the target is the metadata for a class, and that the result is to  
 1998 be returned in CIM-XML format.
- 1999 • (line 10) The ClassName Selector specifies that the class targeted by the Get operation is  
 2000 CIM\_ComputerSystem
- 2001 • (line 11) The \_\_cimnamespace Selector specifies that the class metadata is desired for the  
 2002 root/interop namespace.
- 2003 • (line 14) The IncludeInheritedElements Option (true) specifies to return in the class metadata elements  
 2004 that are inherited from parent classes. Elements may include property definitions, qualifiers, and so  
 2005 forth, depending on the capabilities of the service.
- 2006 • (line 15) The IncludeQualifiers Option (true) specifies to return in the class metadata the qualifiers  
 2007 declared in the MOF that defines the class.
- 2008 • (line 13) The OptionSet specifies mustUnderstand="true". The service must process the OptionSet  
 2009 element or fault the request.
- 2010 • (lines 14-15) The two Options specify MustComply="true". The service must honor the Options or fault  
 2011 the request.

2012 The following XML fragment illustrates the metadata that is returned in response to the Get request.

```

2013 (1) <!-- Example fragment of XML for the data returned due to a get class operation. -->
2014 (2) <env:Envelope>

```

```

2015 (3) <env:Header>
2016 (4)
2017 <wsa04:Action>http://schemas.xmlsoap.org/ws/2004/09/transfer/GetResponse</wsa04:Action>
2018 (5) ...
2019 (6) </env:Header>
2020 (7) <env:Body>
2021 (8) <wsmb:Metadata wsmb:format="CIM-XML"
2022 (9) wsmb:cimClass="http://schemas.dmtf.org/wscim/1/cim-schema/2/CIM_ComputerSystem">
2023 (10) <cim:CLASS SUPERCLASS="CIM_System" NAME="CIM_ComputerSystem">
2024 (11) ...
2025 (12) </cim:CLASS>
2026 (13) </wsmb:Metadata>
2027 (14) </env:Body>
2028 (15) </env:Envelope>

```

2029 The response includes the following elements.

- 2030 • (line 8) The Metadata element contains the class metadata for CIM\_ComputerSystem. It indicates that  
2031 the format of the metadata is CIM-XML. It also specifies the full class name URI for  
2032 CIM\_ComputerSystem.
- 2033 • (line 10) The CLASS element specifies the class metadata for CIM\_ComputerSystem in CIM-XML  
2034 format (as specified in [DSP0201](#) and [DSP0203](#)).
- 2035 • (line 11) An ellipsis indicates that the bulk of the actual metadata text is not included in this and other  
2036 examples. Such metadata is typically long and version-specific.

## 2037 15.14 Example: Enumerate EPRs for Class Metadata for CIM\_ComputerSystem 2038 and Classes Derived from It

2039 The following XML fragment illustrates the use of the ResourceURI, Selectors, and Options to specify an  
2040 operation targeted to retrieve metadata.

```

2041 (1) <!-- Example fragment of XML for an enumerate class operation. -->
2042 (2) <env:Envelope>
2043 (3) <env:Header>
2044 (4) <wsa04:To>http://somedomain.tld:80/wsman</wsa04:To>
2045 (5)
2046 <wsa04:Action>http://schemas.xmlsoap.org/ws/2004/09/enumeration/Enumerate</wsa04:Action>
2047 (6) <wsman:ResourceURI>
2048 (7) http://schemas.dmtf.org/wbem/ws-cim/1/cim-schema/2/*
2049 (8) </wsman:ResourceURI>
2050 (9) <wsman:SelectorSet>
2051 (10) <wsman:Selector Name="ClassName">CIM_ComputerSystem</wsman:Selector>
2052 (11) <wsman:Selector Name="__cimnamespace">root/interop</wsman:Selector>
2053 (12) </wsman:SelectorSet>
2054 (13) <wsman:OptionSet mustUnderstand="true">
2055 (14) <wsman:Option Name="IncludeSubclasses" MustComply="true">true</wsman:Option>
2056 (15) <wsman:Option Name="IncludePathEPR" MustComply="true">true</wsman:Option>
2057 (16) <wsman:Option Name="ExcludeClassSpecification"
2058 MustComply="true">true</wsman:Option>
2059 (17) </wsman:OptionSet>
2060 (18) </env:Header>
2061 (19) <env:Body>
2062 (20) <wsman:Enumerate>
2063 (21) ...

```

```

2064 (22)      <wsman:OptimizeEnumeration />
2065 (23)      <wsman:MaxElements>30</wsman:MaxElements>
2066 (24)      </wsmen:Enumerate>
2067 (25)      </env:Body>
2068 (26)      </env:Envelope>

```

2069 The request includes the following elements.

- 2070 • (line 5) The Action specifies to Enumerate all of the target.
- 2071 • (line 6) The ResourceURI specifies that the target is the metadata for a class, and that the result is to  
2072 be returned in WS-CIM format.
- 2073 • (line 10) The ClassName Selector specifies that the root class of the enumeration is  
2074 CIM\_ComputerSystem
- 2075 • (line 11) The \_\_cimnamespace Selector specifies that the class metadata is desired for the  
2076 root/interop namespace.
- 2077 • (line 14) The IncludeSubclasses Option (true) specifies to return metadata for all classes (in the  
2078 namespace) that are named, or are derived from, CIM\_ComputerSystem.
- 2079 • (line 15) The IncludePathEPR Option (true) specifies to return EPRs to the class definitions. These  
2080 could be used in future Get operations to retrieve the class metadata.
- 2081 • (line 16) The ExcludeClassSpecification Option (true) specifies to not return any metadata or other  
2082 elements describing the definition of the class. Because the IncludePath Option is specified, only the  
2083 EPRs to the class definitions will be returned.
- 2084 • (line 13) The OptionSet specifies mustUnderstand="true". The service must process the OptionSet  
2085 element or fault the request.
- 2086 • (lines 14-16) The several Options specify MustComply="true". The service must honor the Options or  
2087 fault the request.
- 2088 • (line 22) The OptimizeEnumeration specifies that an optimized enumeration should be used to retrieve  
2089 some or all of the results in the initial response.
- 2090 • (line 23) The MaxElements specifies that up to 30 EPRs should be returned in the initial response.

2091 The following XML fragment illustrates the metadata that is returned in response to the Enumeration  
2092 request.

```

2093 (1) <!-- Example fragment of XML for the data returned due to an enumerate class operation.
2094 -->
2095 (2) <env:Envelope>
2096 (3)   <env:Header>
2097 (4)     <wsa04:Action>
2098 (5)       http://schemas.xmlsoap.org/ws/2004/09/enumeration/EnumerateResponse
2099 (6)     </wsa04:Action>
2100 (7)     ...
2101 (8)   </env:Header>
2102 (9)   <env:Body>
2103 (10)    <wsmen:EnumerateResponse>
2104 (11)      <wsmen:EnumerationContext>...</wsmen:EnumerationContext>
2105 (12)      <wsman:Items>
2106 (13)        <wsa04:EndpointReference>
2107 (14)          <wsa04:Address> ... </wsa04:Address>
2108 (15)          <wsa04:ReferenceParameters>
2109 (16)        <wsman:ResourceURI>

```



```

2110 (17)          http://schemas.dmtf.org/wbem/cim-xml/2/cim-schema/2/CIM_ComputerSystem
2111 (18)          </wsman:ResourceURI>
2112 (19)          ...
2113 (20)          </wsa04:ReferenceParameters>
2114 (21)          </wsa04:EndpointReference>
2115 (22)          <wsa04:EndpointReference>
2116 (23)          <wsa04:Address> ... </wsa04:Address>
2117 (24)          <wsa04:ReferenceParameters>
2118 (25)          <wsman:ResourceURI>
2119 (26)          http://schemas.mycompany.com/wbem/cim-xml/2/cim-
2120 schema/2/My_ComputerSystem
2121 (27)          </wsman:ResourceURI>
2122 (28)          ...
2123 (29)          </wsa04:ReferenceParameters>
2124 (30)          </wsa04:EndpointReference>
2125 (31)          ...
2126 (32)          </wsman:Items>
2127 (33)          </wsman:EndOfSequence />
2128 (34)          </wsmen:EnumerateResponse>
2129 (35)          </env:Body>
2130 (36)          </env:Envelope>

```

2131 The response includes the following elements.

- 2132 • (lines 13-21) The EndpointReference specifies an EPR that can be used in a Get operation to retrieve  
2133 the class metadata for CIM\_ComputerSystem.
- 2134 • (lines 22-30) The EndpointReference specifies an EPR that can be used in a Get operation to retrieve  
2135 the class metadata for My\_ComputerSystem, which derives from CIM\_ComputerSystem.
- 2136 • (line 33) The EndOfSequence specifies that there are no more EPRs to be retrieved.

## 2137 15.15 Example: Enumerate WS-CIM Class Metadata for CIM\_ComputerSystem and 2138 Classes Derived from It

2139 The following XML fragment illustrates the use of the ResourceURI, Selectors, and Options to specify an  
2140 operation targeted to retrieve metadata.

```

2141 (1) <!-- Example fragment of XML for an enumerate class operation. -->
2142 (2) <env:Envelope>
2143 (3)   <env:Header>
2144 (4)     <wsa04:To>http://somedomain.tld:80/wsman</wsa04:To>
2145 (5)
2146 (6)   <wsa04:Action>http://schemas.xmlsoap.org/ws/2004/09/enumeration/Enumerate</wsa04:Action>
2147 (7)     <wsman:ResourceURI>
2148 (8)       http://schemas.dmtf.org/wbem/ws-cim/1/cim-schema/2/*
2149 (9)     </wsman:ResourceURI>
2150 (10)    <wsman:SelectorSet>
2151 (11)      <wsman:Selector Name="ClassName">CIM_ComputerSystem</wsman:Selector>
2152 (12)      <wsman:Selector Name="__cimnamespace">root/interop</wsman:Selector>
2153 (13)    </wsman:SelectorSet>
2154 (14)    <wsman:OptionSet mustUnderstand="true">
2155 (15)      <wsman:Option Name="IncludeSubclasses" MustComply="true">true</wsman:Option>
2156 (16)      <wsman:Option Name="IncludePath" MustComply="true">false</wsman:Option>
2157 (17)    </wsman:OptionSet>

```

```

2158 (17)    </env:Header>
2159 (18)    <env:Body>
2160 (19)    </env:Body>
2161 (20)    </env:Envelope>

```

2162 The request includes the following elements.

- 2163 • (line 5) The Action specifies to Enumerate all of the target.
- 2164 • (line 6) The ResourceURI specifies that the target is the metadata for a class, and that the result is to  
2165 be returned in WS-CIM format.
- 2166 • (line 10) The ClassName Selector specifies that the root class of the enumeration is  
2167 CIM\_ComputerSystem
- 2168 • (line 11) The \_\_cimnamespace Selector specifies that the class metadata is desired for the  
2169 root/interop namespace.
- 2170 • (line 14) The IncludeSubclasses Option (true) specifies to return metadata for all classes (in the  
2171 namespace) that are named, or are derived from, CIM\_ComputerSystem.
- 2172 • (line 15) The IncludePath Option (false) specifies to not return EPRs to the class definitions. Only the  
2173 class metadata will be returned.
- 2174 • (line 13) The OptionSet specifies mustUnderstand="true". The service must process the OptionSet  
2175 element or fault the request.
- 2176 • (lines 14-15) The two Options specify MustComply="true". The service must honor the Options or fault  
2177 the request.

2178 The following XML fragment illustrates the metadata that is returned in response to subsequent Pull  
2179 requests.

```

2180 (1) <!-- Example fragment of XML for the data returned due to an enumerate class operation.
2181 -->
2182 (2) <env:Envelope>
2183 (3)   <env:Header>
2184 (4)     <wsa04:Action>http://schemas.xmlsoap.org/ws/2004/09/enumeration/PullResponse</wsa04:Actio
2185     n>
2186 (5)     ...
2187 (6)   </env:Header>
2188 (7)   <env:Body>
2189 (8)     <wsmen:PullResponse ...>
2190 (9)       <wsmen:EnumerationContext>...</wsmen:EnumerationContext>
2191 (10)      <wsmen:Items>
2192 (11)        <wsmb:Metadata wsmb:format="WS-CIM" wsmb:cimClass=
2193 (12)          "http://schemas.dmtf.org/wscim/1/cim-schema/2/CIM_ComputerSystem">
2194 (13)          <xs:schema targetNamespace=
2195 (14)            "http://schemas.dmtf.org/wscim/1/cim-schema/2/CIM_ComputerSystem" ...>
2196 (15)            ...
2197 (16)          </xs:schema>
2198 (17)        </wsmb:Metadata>
2199 (18)        <wsmb:Metadata wsmb:format="WS-CIM" wsmb:cimClass=
2200 (19)          "http://schemas.mycompany.com/wscim/1/cim-schema/2/My_ComputerSystem">
2201 (20)          <xs:schema targetNamespace=
2202 (21)            "http://schemas.mycompany.com/wscim/1/cim-schema/2/My_ComputerSystem" ...>
2203 (22)            ...
2204 (23)          </xs:schema>

```

```

2206 (24)         </wsmb:Metadata>
2207 (25)         ...
2208 (26)         </wsmen:Items>
2209 (27)         </wsmen:PullResponse>
2210 (28)         </env:Body>
2211 (29)         </env:Envelope>

```

2212 The response includes the following elements.

- 2213 • (line 11) The Metadata element contains the class metadata for CIM\_ComputerSystem. It indicates  
2214 that the format of the metadata is WS-CIM. It also specifies the full class name URI for  
2215 CIM\_ComputerSystem.
- 2216 • (line 13) The schema element specifies the class metadata for CIM\_ComputerSystem in WS-CIM  
2217 format (as specified in [DSP0230](#)).
- 2218 • (line 18) The Metadata element contains the class metadata for My\_ComputerSystem. It indicates that  
2219 the format of the metadata is CIM-XML. It also specifies the full class name URI for  
2220 My\_ComputerSystem.
- 2221 • (line 20) The schema element specifies the class metadata for My\_ComputerSystem in WS-CIM  
2222 format (as specified in [DSP0230](#)).

## 2223 15.16 Example: Enumerate CIM-XML Class Metadata and EPRs for 2224 CIM\_ComputerSystem and Classes Derived from It

2225 The following XML fragment illustrates the use of the ResourceURI, Selectors, and Options to specify an  
2226 operation targeted to retrieve metadata.

```

2227 (1) <!-- Example fragment of XML for an enumerate class operation. -->
2228 (2) <env:Envelope>
2229 (3)   <env:Header>
2230 (4)     <wsa04:To>http://somedomain.tld:80/wsman</wsa04:To>
2231 (5)     <wsa04:Action>http://schemas.xmlsoap.org/ws/2004/09/enumeration/Enumerate</wsa04:Action>
2232 (6)     <wsman:ResourceURI>
2233 (7)       http://schemas.dmtf.org/wbem/cim-xml/2/cim-schema/2/*
2234 (8)     </wsman:ResourceURI>
2235 (9)     <wsman:SelectorSet>
2236 (10)      <wsman:Selector Name="ClassName">CIM_ComputerSystem</wsman:Selector>
2237 (11)      <wsman:Selector Name="__cimnamespace">root/interop</wsman:Selector>
2238 (12)    </wsman:SelectorSet>
2239 (13)    <wsman:OptionSet mustUnderstand="true">
2240 (14)      <wsman:Option Name="IncludeSubclasses MustComply="true">true</wsman:Option>
2241 (15)      <wsman:Option Name="IncludePath MustComply="true">true</wsman:Option>
2242 (16)      <wsman:Option Name="IncludeInheritedElements"
2243 (16)      MustComply="true">true</wsman:Option>
2244 (17)    </wsman:OptionSet>
2245 (18)  </env:Header>
2246 (19)  <env:Body>
2247 (20)  </env:Body>
2248 (21) </env:Envelope>

```

2250 The request includes the following elements.

- 2251 • (line 5) The Action specifies to Enumerate all of the target.

- 2252 • (line 6) The ResourceURI specifies that the target is the metadata for a class, and that the result is to  
2253 be returned in CIM-XML format.
- 2254 • (line 10) The ClassName Selector specifies that the root class of the enumeration is  
2255 CIM\_ComputerSystem
- 2256 • (line 11) The \_\_cimnamespace Selector specifies that the class metadata is desired for the  
2257 root/interop namespace.
- 2258 • (line 14) The IncludeSubclasses Option (true) specifies to return metadata for all classes (in the  
2259 namespace) that are named, or are derived from, CIM\_ComputerSystem.
- 2260 • (line 15) The IncludePath Option (true) specifies to return EPRs to the class definitions. These could  
2261 be used in future Get operations to retrieve the class metadata.
- 2262 • (line 16) The IncludeInheritedElements Option (true) specifies to return in the class metadata elements  
2263 that are inherited from parent classes. Elements may include property definitions, qualifiers, and so  
2264 forth, depending on the capabilities of the service.
- 2265 • (line 13) The OptionSet specifies mustUnderstand="true". The service must process the OptionSet  
2266 element or fault the request.
- 2267 • (lines 14-16) The several Options specify MustComply="true". The service must honor the Options or  
2268 fault the request.

2269 The following XML fragment illustrates the metadata that is returned in response to subsequent Pull  
2270 requests.

```

2271 (1) <!-- Example fragment of XML for the data returned due to an enumerate class operation.
2272 -->
2273 (2) <env:Envelope>
2274 (3)   <env:Header>
2275 (4)     <wsa04:Action>http://schemas.xmlsoap.org/ws/2004/09/enumeration/PullResponse</wsa04:Actio
2276     n>
2277 (5)     ...
2278 (6)   </env:Header>
2279 (7)   <env:Body>
2280 (8)     <wsmen:PullResponse ...>
2281 (9)       <wsmen:EnumerationContext>...</wsmen:EnumerationContext>
2282 (10)      <wsmen:Items>
2283 (11)        <wsman:Item>
2284 (12)          <wsa04:EndpointReference>
2285 (13)            <wsa04:Address> ... </wsa04:Address>
2286 (14)            <wsa04:ReferenceParameters>
2287 (15)              <wsman:ResourceURI>
2288 (16)                http://schemas.dmtf.org/wbem/cim-xml/2/cim-schema/2/CIM_ComputerSystem
2289 (17)              </wsman:ResourceURI>
2290 (18)              ...
2291 (19)            </wsa04:ReferenceParameters>
2292 (20)          </wsa04:EndpointReference>
2293 (21)          <wsmb:Metadata wsmb:format="CIM-XML" wsmb:cimClass=
2294 (22)            "http://schemas.dmtf.org/wscim/1/cim-schema/2/CIM_ComputerSystem">
2295 (23)            <cim:CLASS SUPERCLASS="CIM_System" NAME="CIM_ComputerSystem">
2296 (24)              ...
2297 (25)            </cim:CLASS>
2298 (26)          </wsmb:Metadata>
2299 (27)        </wsman:Item>
2300

```

```

2301 (28)      <wsman:Item>
2302 (29)      <wsa04:EndpointReference>
2303 (30)      <wsa04:Address> ... </wsa04:Address>
2304 (31)      <wsa04:ReferenceParameters>
2305 (32)      <wsman:ResourceURI>
2306 (33)      http://schemas.mycompany.com/wbem/cim-xml/2/cim-
2307 schema/2/My_ComputerSystem
2308 (34)      </wsman:ResourceURI>
2309 (35)      ...
2310 (36)      </wsa04:ReferenceParameters>
2311 (37)      </wsa04:EndpointReference>
2312 (38)      <wsmb:Metadata wsmb:format="CIM-XML" wsmb:cimClass=
2313 (39)      "http://schemas.mycompany.com/wscim/1/cim-schema/2/My_ComputerSystem">
2314 (40)      <cim:CLASS SUPERCLASS="CIM_ComputerSystem" NAME="My_ComputerSystem">
2315 (41)      ...
2316 (42)      </cim:CLASS>
2317 (43)      </wsmb:Metadata>
2318 (44)      </wsman:Item>
2319 (45)      ...
2320 (46)      </wsman:Items>
2321 (47)      </wsman:PullResponse>
2322 (48)      </env:Body>
2323 (49)      </env:Envelope>

```

2324 The response includes the following elements.

- 2325 • (line 11) The Item element contains the EPR and class metadata for one of the returned classes,  
2326 CIM\_ComputerSystem.
- 2327 • (lines 12-20) The EndpointReference specifies an EPR that can be used in a Get operation to retrieve  
2328 the class metadata for CIM\_ComputerSystem.
- 2329 • (line 21) The Metadata element contains the class metadata for CIM\_ComputerSystem. It indicates  
2330 that the format of the metadata is CIM-XML. It also specifies the full class name URI for  
2331 CIM\_ComputerSystem.
- 2332 • (line 23) The CLASS element specifies the class metadata for CIM\_ComputerSystem in CIM-XML  
2333 format (as specified in [DSP0201](#) and [DSP0203](#)).
- 2334 • (line 28) The Item element contains the EPR and class metadata for another one of the returned  
2335 classes that derives from CIM\_ComputerSystem, My\_ComputerSystem.
- 2336 • (lines 29-37) The EndpointReference specifies an EPR that can be used in a Get operation to retrieve  
2337 the class metadata for My\_ComputerSystem.
- 2338 • (line 38) The Metadata element contains the class metadata for My\_ComputerSystem. It indicates that  
2339 the format of the metadata is CIM-XML. It also specifies the full class name URI for  
2340 My\_ComputerSystem.
- 2341 • (line 40) The CLASS element specifies the class metadata for My\_ComputerSystem in CIM-XML  
2342 format (as specified in [DSP0201](#) and [DSP0203](#)).

## 2343 16 Fault Codes

2344 Faults defined in this specification must use the following action URI:

2345 <http://schemas.dmtf.org/wbem/wsman/1/cimbinding/fault>

2346 **16.1 wsmb:CIMException**

2347 Table 12 provides information about the wsmb:CIMException fault subcode.

2348 **Table 12 – wsmb:CIMException**

Fault Subcode	wsmb:CIMException
Action URI	http://schemas.dmtf.org/wbem/wsman/1/cimbinding/fault
Code	s:Receiver
Reason	The CIM server encountered an exception during the processing of the request.
Detail	XML representation of CIM_Error instance
Comments	
Applicability	Any message
Remedy	Depends upon the exception

2349 **16.2 wsmb:PolymorphismModeNotSupported**

2350 Table 13 provides information about the wsmb:PolymorphismModeNotSupported fault subcode.

2351 **Table 13 – wsmb:PolymorphismModeNotSupported**

Fault Subcode	wsmb:PolymorphismModeNotSupported
Action URI	http://schemas.dmtf.org/wbem/wsman/1/cimbinding/fault
Code	s:Sender
Reason	The resource does not support the requested polymorphism mode.
Detail	
Comments	
Applicability	wsen:Enumerate, wse:Subscribe
Remedy	Try the request again without specifying a polymorphism mode.

2352 **17 Mapping for DSP0200 CIM Operations**

2353 CIM Profiles define support for CIM operations for each CIM class used in the profile. These supported  
 2354 operations are defined in [DSP0200](#). This clause outlines the WS-Management equivalent operations for  
 2355 each supported CIM operation that is defined in [DSP0200](#) and additional uses of WS-Management  
 2356 functionality to achieve the same goal.

2357 **17.1 Supported Operations**

2358 The following CIM operations have equivalents defined by this specification:

- 2359 • GetInstance: This operation is used to return a single CIM instance from the target  
2360 namespace.
- 2361 • DeleteInstance: This operation is used to delete a single CIM instance from the target  
2362 namespace.
- 2363 • ModifyInstance: This operation is used to modify a single CIM instance in the target  
2364 namespace.
- 2365 • CreateInstance: This operation is used to create a single CIM instance in the target  
2366 namespace.

- 2367 • EnumerateInstances: This operation is used to enumerate instances of a CIM Class (this  
2368 includes instances in the class and any subclasses in accordance with the polymorphic  
2369 nature of CIM objects) in the target Namespace.
- 2370 • EnumerateInstanceNames: This operation is used to enumerate the names (model paths) of  
2371 the instances of a CIM Class (this includes instances in the class and any subclasses in  
2372 accordance with the polymorphic nature of CIM objects) in the target Namespace.
- 2373 • Associators: This operation is used to enumerate CIM Objects (Classes or Instances) that  
2374 are associated to a particular source CIM Object.
- 2375 • AssociatorsNames: This operation is used to enumerate the names of CIM Objects (Classes  
2376 or Instances) that are associated to a particular source CIM Object.
- 2377 • References: This operation is used to enumerate the association objects that refer to a  
2378 particular target CIM Object (Class or Instance).
- 2379 • ReferenceNames: This operation is used to enumerate the association objects that refer to a  
2380 particular target CIM Object (Class or Instance).

2381 The following subclauses define the mapping of the above operations over WS-Management.

2382 **17.1.1 GetInstance**

2383 The mapping defined in Table 14 shall be used for the GetInstance operation.

2384 **Table 14 – GetInstance**

Operation	GetInstance
Operation target	CIM Server
WS-Man operation	WS-Management 1.1 resource access Get
EPR	Class-specific ResourceURI with keys as selectors
Additional usage	None
Notes	Can be targeted only at the class of the actual instance

2385 Table 15 provides the mapping of GetInstance arguments defined in clause 5.3.2.2 of [DSP0200](#).

2386 **Table 15 – GetInstance Arguments**

Argument	GetInstance
InstanceName	Mapped to EPR
LocalOnly	false
IncludeQualifier	false
IncludeClassOrigin	false
PropertyList[ ]	If it is NULL, then the operation is handled through WS-Management 1.1 resource access Get. If it is not NULL, then the operation is handled through fragment level WS-Management 1.1 resource access Get (see clause 7.8 of <a href="#">DSP0226</a> ).

2387 Table 16 provides the mapping of status codes defined in [DSP0200](#) to equivalent SOAP faults defined in  
2388 [DSP0226](#).

2389 **Table 16 – GetInstance Error Codes**

Status Code	Equivalent SOAP Fault
CIM_ERR_ACCESS_DENIED	wsman:AccessDenied
CIM_ERR_INVALID_NAMESPACE	wsa:DestinationUnreachable

CIM_ERR_INVALID_PARAMETER	wsman:InvalidParameter
CIM_ERR_INVALID_CLASS	wsa:DestinationUnreachable
CIM_ERR_NOT_FOUND	wsa:DestinationUnreachable
CIM_ERR_FAILED	wsman:InternalError

### 2390 17.1.2 DeleteInstance

2391 The mapping defined in Table 17 shall be used for the DeleteInstance operation.

2392 **Table 17 – DeleteInstance**

Operation	DeleteInstance
Operation target	CIM Server
WS-Man operation	WS-Management 1.1 resource access Delete or WS-Management 1.1 notifications Unsubscribe (for CIM_IndicationSubscription and CIM_FilterCollectionSubscription)
EPR	Class-specific ResourceURI with keys as selectors
Additional usage	None

2393 Table 18 provides the mapping of the DeleteInstance arguments defined in clause 5.3.2.4 of [DSP0200](#).

2394 **Table 18 – DeleteInstance Arguments**

Argument	DeleteInstance
InstanceName	Mapped to EPR

2395 Table 19 provides the mapping of status codes defined in [DSP0200](#) to equivalent SOAP faults defined in  
2396 [DSP0226](#).

2397 **Table 19 – DeleteInstance Error Codes**

Status Code	Equivalent SOAP Fault
CIM_ERR_ACCESS_DENIED	wsman:AccessDenied
CIM_ERR_NOT_SUPPORTED (by the CIM Server for this operation)	wsa:ActionNotSupported
CIM_ERR_INVALID_NAMESPACE	wsa:DestinationUnreachable
CIM_ERR_INVALID_PARAMETER	wsman:InvalidParameter
CIM_ERR_INVALID_CLASS	wsa:DestinationUnreachable
CIM_ERR_NOT_SUPPORTED (This operation is not supported for the class of the specified instance, if provided.)	wsa:ActionNotSupported
CIM_ERR_NOT_FOUND	wsa:DestinationUnreachable
CIM_ERR_FAILED	wsman:InternalError

### 2398 17.1.3 ModifyInstance

2399 The mapping defined in Table 20 shall be used for the ModifyInstance operation.



2400

**Table 20 – ModifyInstance**

Operation	ModifyInstance
Operation target	CIM Server
WS-Man operation	WS-Management 1.1 resource access Put or WS-Management 1.1 notifications Renew (for CIM_IndicationSubscription and CIM_FilterCollectionSubscription)
EPR	Class-specific ResourceURI with keys as selectors
Additional usage	None
Notes	Can be targeted only at the class of the actual instance

2401 Table 21 provides the mapping of the ModifyInstance arguments defined in clause 5.3.2.8 of [DSP0200](#).

2402

**Table 21 – ModifyInstance Arguments**

Argument	ModifyInstance
InstanceName	Mapped to EPR
IncludeQualifier	false
PropertyList[ ]	Always set to NULL for the instances of CIM_IndicationSubscription and CIM_FilterCollectionSubscription. For instances of other classes: If it is NULL, then the operation is handled through WS-Management 1.1 resource access Put. If it is not NULL, then the operation is handled through fragment level WS-Management 1.1 resource access Put (clause 7.9 of <a href="#">DSP0226</a> ).

2403 Table 22 provides the mapping of status codes defined in [DSP0200](#) to equivalent SOAP faults defined in  
2404 [DSP0226](#).

2405

**Table 22 – ModifyInstance Error Codes**

Status Code	Equivalent SOAP Fault
CIM_ERR_ACCESS_DENIED	wsman:AccessDenied
CIM_ERR_NOT_SUPPORTED (by the CIM Server for this operation)	wsa:ActionNotSupported
CIM_ERR_INVALID_NAMESPACE	wsa:DestinationUnreachable
CIM_ERR_INVALID_PARAMETER	wsman:InvalidParameter
CIM_ERR_INVALID_CLASS	wsa:DestinationUnreachable
CIM_ERR_NOT_SUPPORTED (This operation is not supported for the class of the specified instance, if provided.)	wsa:ActionNotSupported
CIM_ERR_NOT_FOUND	wsa:DestinationUnreachable
CIM_ERR_FAILED	wsman:InternalError

2406 **17.1.4 CreateInstance**

2407 The mapping defined in Table 23 shall be used for the CreateInstance operation.

2408

**Table 23 – CreateInstance**

Operation	CreateInstance
Operation target	CIM Server
WS-Man operation	WS-Management 1.1 resource access Create or WS-Management 1.1 notifications Subscribe (for CIM_IndicationSubscription and CIM_FilterCollectionSubscription)
EPR	Class-specific ResourceURI as factory, with only the __cimnamespace selector allowed
Additional usage	None
Notes	Can be targeted only at the class of actual instance

2409 Table 24 provides the mapping of the CreateInstance arguments as defined in clause 5.3.2.6 of [DSP0200](#).

2410

**Table 24 – CreateInstance Arguments**

Argument	CreateInstance
InstanceName	Mapped to EPR

2411 Table 25 provides the mapping of status codes defined in [DSP0200](#) to equivalent SOAP faults defined in [DSP0226](#).

2413

**Table 25 – CreateInstance Error Codes**

Status Code	Equivalent SOAP Fault
CIM_ERR_ACCESS_DENIED	wsman:AccessDenied
CIM_ERR_NOT_SUPPORTED (by the CIM Server for this operation)	wsa:ActionNotSupported
CIM_ERR_INVALID_NAMESPACE	wsa:DestinationUnreachable
CIM_ERR_INVALID_PARAMETER	wsman:InvalidParameter
CIM_ERR_INVALID_CLASS	wsa:DestinationUnreachable
CIM_ERR_NOT_SUPPORTED (This operation is not supported for the class of the specified instance, if provided.)	wsa:ActionNotSupported
CIM_ERR_ALREADY_EXISTS	wsman:AlreadyExists
CIM_ERR_FAILED	wsman:InternalError

2414 **17.1.5 EnumerateInstances**

2415 The mapping defined in Table 26 shall be used for the EnumerateInstances operation.

2416

**Table 26 – EnumerateInstances**

Operation	EnumerateInstances
Operation target	CIM Server
WS-Man operation	WS-Management 1.1 Enumerate
EPR	Class-specific ResourceURI with no selectors
Additional usage	Use wsman:EnumerationMode=EnumerateObjectAndEPR
Notes	

2417 Table 27 provides the mapping of EnumerateInstances arguments as defined in clause 5.3.2.11 of  
 2418 [DSP0200](#).

2419 **Table 27 – EnumerateInstances Arguments**

Argument	EnumerateInstances
ClassName	Mapped to EPR
LocalOnly	false
DeepInheritance	If true, then wsmb:PolymorphismMode modifier element value is set to IncludeSubClassProperties or wsmb:PolymorphismMode is not specified. If false, then wsmb:PolymorphismMode modifier element value is set to ExcludeSubClassProperties.
IncludeQualifier	false
IncludeClassOrigin	false
PropertyList[ ]	If it is NULL, then the operation is handled through WS-Management 1.1 Enumeration. If it is not NULL, then the operation is handled through fragment-level enumerations (see clause 8.6 of <a href="#">DSP0226</a> ).

2420 Table 28 provides the mapping of status codes defined in [DSP0200](#) to equivalent SOAP faults defined in  
 2421 [DSP0226](#).

2422 **Table 28 – EnumerateInstances Error Codes**

Status Code	Equivalent SOAP Fault
CIM_ERR_ACCESS_DENIED	wsman:AccessDenied
CIM_ERR_NOT_SUPPORTED (by the CIM Server for this operation)	wsa:ActionNotSupported
CIM_ERR_INVALID_NAMESPACE	wsa:DestinationUnreachable
CIM_ERR_INVALID_PARAMETER	wsman:InvalidParameter
CIM_ERR_INVALID_CLASS	wsa:DestinationUnreachable
CIM_ERR_NOT_SUPPORTED (This operation is not supported for the class of the specified instance, if provided.)	wsa:ActionNotSupported
CIM_ERR_FAILED	wsman:InternalError

2423 **17.1.6 EnumerateInstanceNames**

2424 The mapping defined in Table 29 shall be used for the EnumerateInstanceNames operation.

2425 **Table 29 – EnumerateInstanceNames**

Operation	EnumerateInstanceNames
Operation target	CIM Server
WS-Man operation	WS-Management 1.1 Enumerate
EPR	Class-specific ResourceURI with no selectors
Additional usage	Use wsman:EnumerationMode=EnumerateEPR
Notes	

2426 Table 30 provides the mapping of EnumerateInstanceNames arguments as defined in clause 5.3.2.12 of  
 2427 [DSP0200](#).

2428 **Table 30 – EnumerateInstanceNames Arguments**

Argument	EnumerateInstanceNames
ClassName	Mapped to EPR

2429 Table 31 provides the mapping of status codes defined in [DSP0200](#) to equivalent SOAP faults defined in  
 2430 [DSP0226](#).

2431 **Table 31 – EnumerateInstanceNames Error Codes**

Status Code	Equivalent SOAP Fault
CIM_ERR_ACCESS_DENIED	wsman:AccessDenied
CIM_ERR_NOT_SUPPORTED (by the CIM Server for this operation)	wsa:ActionNotSupported
CIM_ERR_INVALID_NAMESPACE	wsa:DestinationUnreachable
CIM_ERR_INVALID_PARAMETER	wsman:InvalidParameter
CIM_ERR_INVALID_CLASS	wsa:DestinationUnreachable
CIM_ERR_NOT_SUPPORTED (This operation is not supported for the class of the specified instance, if provided.)	wsa:ActionNotSupported
CIM_ERR_FAILED	wsman:InternalError

2432 **17.1.7 Associators**

2433 The mapping defined in Table 32 shall be used for the Associators operation.

2434 **Table 32 – Associators**

Operation	Associators
Operation target	CIM Server
WS-Man operation	WS-Management 1.1 Enumerate
EPR	All-classes ResourceURI with no selectors
Additional usage	Use wsman:EnumerationMode=EnumerateObjectAndEPR Use the following association filter dialect with the wsmb:AssociatedInstances element: <a href="http://schemas.dmtf.org/wbem/wsman/1/cimbinding/associationFilter">http://schemas.dmtf.org/wbem/wsman/1/cimbinding/associationFilter</a>
Notes	

2435 Table 33 provides the mapping of the Associators arguments as defined in clause 5.3.2.14 of [DSP0200](#).

2436 **Table 33 – Associators Arguments**

Argument	Associators
ObjectName	wsmb:Object value is set to ObjectName
AssocClass	If not NULL, wsmb:AssociationClassName value is set to AssocClass
ResultClass	If not NULL, wsmb:ResultClassName value is set to ResultClass
Role	If not NULL, wsmb:Role value is set to Role

ResultRole	If not NULL, wsmb:ResultRole value is set to ResultRole
IncludeQualifiers	false
IncludeClassOrigin	false
PropertyList[ ]	If it is NULL, then the operation is handled through WS-Management 1.1 Enumeration. If it is not NULL, then the operation is handled through fragment-level enumerations (see clause 8.6 of <a href="#">DSP0226</a> ).

2437 Table 34 provides the mapping of status codes defined in [DSP0200](#) to equivalent SOAP faults defined in  
 2438 [DSP0226](#).

2439 **Table 34 – Associators Error Codes**

Status Code	Equivalent SOAP Fault
CIM_ERR_ACCESS_DENIED	wsman:AccessDenied
CIM_ERR_NOT_SUPPORTED (by the CIM Server for this operation)	wsa:ActionNotSupported
CIM_ERR_INVALID_NAMESPACE	wsa:DestinationUnreachable
CIM_ERR_INVALID_PARAMETER	wsman:InvalidParameter
CIM_ERR_NOT_SUPPORTED (This operation is not supported for the class of the specified instance, if provided.)	wsa:ActionNotSupported
CIM_ERR_FAILED	wsman:InternalError

2440 **17.1.8 AssociatorNames**

2441 The mapping defined in Table 35 shall be used for the AssociatorNames operation.

2442 **Table 35 – AssociatorNames**

Operation	AssociatorNames
Operation target	CIM Server
WS-Man operation	WS-Management 1.1 Enumerate
EPR	All-classes ResourceURI with no selectors
Additional usage	Use wsman:EnumerationMode=EnumerateEPR Use the following association filter dialect with the wsmb:AssociatedInstances element: <a href="http://schemas.dmf.org/wbem/wsman/1/cimbinding/associationFilter">http://schemas.dmf.org/wbem/wsman/1/cimbinding/associationFilter</a>
Notes	

2443 Table 36 provides the mapping of the AssociatorNames arguments as defined in clause 5.3.2.15 of  
 2444 [DSP0200](#).

2445 **Table 36 – AssociatorNames Arguments**

Argument	AssociatorNames
ObjectName	wsmb:Object value is set to ObjectName
AssocClass	If not NULL, wsmb:AssociationClassName value is set to AssocClass
ResultClass	If not NULL, wsmb:ResultClassName value is set to ResultClass
Role	If not NULL, wsmb:Role value is set to Role
ResultRole	If not NULL, wsmb:ResultRole value is set to ResultRole

2446 Table 37 provides the mapping of status codes as defined in [DSP0200](#) to equivalent SOAP faults defined  
 2447 in [DSP0226](#).

2448 **Table 37 – AssociatorNames Error Codes**

Status Code	Equivalent SOAP Fault
CIM_ERR_ACCESS_DENIED	wsman:AccessDenied
CIM_ERR_NOT_SUPPORTED (by the CIM Server for this operation)	wsa:ActionNotSupported
CIM_ERR_INVALID_NAMESPACE	wsa:DestinationUnreachable
CIM_ERR_INVALID_PARAMETER	wsman:InvalidParameter
CIM_ERR_NOT_SUPPORTED (This operation is not supported for the class of the specified instance, if provided.)	wsa:ActionNotSupported
CIM_ERR_FAILED	wsman:InternalError

2449 **17.1.9 References**

2450 The mapping defined in Table 38 shall be used for the References operation.

2451 **Table 38 – References**

Operation	References
Operation target	CIM Server
WS-Man operation	WS-Management 1.1 Enumerate
EPR	All-classes ResourceURI with no selectors
Additional usage	Use wsman:EnumerationMode=EnumerateObjectAndEPR Use association the following filter dialect with the wsmb:AssociationInstances element: <a href="http://schemas.dmtf.org/wbem/wsman/1/cimbinding/associationFilter">http://schemas.dmtf.org/wbem/wsman/1/cimbinding/associationFilter</a>
Notes	

2452 Table 39 provides the mapping of the References arguments as defined in clause 5.3.2.16 of [DSP0200](#).

2453 **Table 39 – References Arguments**

Argument	References
ObjectName	wsmb:Object value is set to ObjectName
ResultClass	If not NULL, wsmb:ResultClassName value is set to ResultClass
Role	If not NULL, wsmb:Role value is set to Role
IncludeQualifiers	false
IncludeClassOrigin	false
PropertyList[ ]	If it is NULL, then the operation is handled through WS-Management 1.1 Enumeration. If it is not NULL, then the operation is handled through fragment-level enumerations (see clause 8.6 of <a href="#">DSP0226</a> ).

2454 Table 40 provides the mapping of status codes as defined in [DSP0200](#) to equivalent SOAP faults defined  
 2455 in [DSP0226](#).

2456

**Table 40 – References Error Codes**

Status Code	Equivalent SOAP Fault
CIM_ERR_ACCESS_DENIED	wsman:AccessDenied
CIM_ERR_NOT_SUPPORTED (by the CIM Server for this operation)	wsa:ActionNotSupported
CIM_ERR_INVALID_NAMESPACE	wsa:DestinationUnreachable
CIM_ERR_INVALID_PARAMETER	wsman:InvalidParameter
CIM_ERR_NOT_SUPPORTED (This operation is not supported for the class of the specified instance, if provided.)	wsa:ActionNotSupported
CIM_ERR_FAILED	wsman:InternalError

2457 **17.1.10 ReferenceNames**

2458 The mapping defined in Table 41 shall be used for the ReferenceNames operation.

2459

**Table 41 – ReferenceNames**

Operation	ReferenceNames
Operation target	CIM Server
WS-Man operation	WS-Management 1.1 Enumerate
EPR	All-classes ResourceURI with no selectors
Additional usage	Use wsman:EnumerationMode=EnumerateEPR Use association the following filter dialect with the wsmb:AssociationInstances element: <a href="http://schemas.dmtf.org/wbem/wsman/1/cimbinding/associationFilter">http://schemas.dmtf.org/wbem/wsman/1/cimbinding/associationFilter</a>
Notes	

2460 Table 42 provides the mapping of the ReferenceNames arguments as defined in clause 5.3.2.17 of  
2461 [DSP0200](#).

2462

**Table 42 – ReferenceNames Arguments**

Argument	ReferenceNames
ObjectName	wsmb:Object value is set to ObjectName
ResultClass	If not NULL, wsmb:ResultClassName value is set to ResultClass
Role	If not NULL, wsmb:Role value is set to Role

2463 Table 43 provides the mapping of status codes as defined in [DSP0200](#) to equivalent SOAP faults defined  
2464 in [DSP0226](#).

2465

**Table 43 – ReferenceNames Error Codes**

Status Code	Equivalent SOAP Fault
CIM_ERR_ACCESS_DENIED	wsman:AccessDenied
CIM_ERR_NOT_SUPPORTED (by the CIM Server for this operation)	wsa:ActionNotSupported
CIM_ERR_INVALID_NAMESPACE	wsa:DestinationUnreachable
CIM_ERR_INVALID_PARAMETER	wsman:InvalidParameter

CIM_ERR_NOT_SUPPORTED (This operation is not supported for the class of the specified instance, if provided.)	wsa:ActionNotSupported
CIM_ERR_FAILED	wsman:InternalError

2466 **17.1.11 ExecQuery**

2467 This operation is supported for the CIM query language (CQL). See 8.1 for more details.

2468 **17.2 Unsupported Operations**

2469 This specification does not define equivalents for the following operations:

- 2470 • GetClass
- 2471 • DeleteClass
- 2472 • CreateClass
- 2473 • ModifyClass
- 2474 • EnumerateClasses
- 2475 • EnumerateClassNames
- 2476 • GetProperty
- 2477 • SetProperty
- 2478 • GetQualifier
- 2479 • SetQualifier
- 2480 • DeleteQualifier
- 2481 • EnumerateQualifiers

2482 **18 Mapping of Error Messages to SOAP Fault Subcodes**

2483 Table 44 outlines suggested mappings of CIM error messages to corresponding subcodes to be used when  
 2484 returning SOAP faults.

2485 **Table 44 – CIM Error Messages with Corresponding Subcode Mappings**

Message ID	Message Name	Fault Subcode
WIPG0201	Authentication failed	wsman:AccessDenied (Support may be transport-dependent.)
WIPG0202	Authorization failed	wsman:AccessDenied
WIPG0203	Operation not supported by CIM service infrastructure	wsa:ActionNotSupported
WIPG0204	CIM namespace not found	wsa:DestinationUnreachable
WIPG0205	Missing input parameter	wsmb:CIMException
WIPG0206	Duplicate input parameter	wsman:InvalidParameter
WIPG0207	Unknown input parameter	wsman:InvalidParameter



Message ID	Message Name	Fault Subcode
WIPG0208	Invalid input parameter value	wsman:InvalidParameter
WIPG0213	CIM instance not found	wsa:DestinationUnreachable
WIPG0214	CIM class not found	wsa:DestinationUnreachable
WIPG0216	CIM instance already exists	wsman:AlreadyExists
WIPG0218	No such CIM method	wsa:ActionNotSupported
WIPG0219	CIM method not supported by CIM class implementation	wsa:ActionNotSupported
WIPG0220	No such CIM property	wxf:InvalidRepresentation
WIPG0221	Unknown query language	wsen:FilterDialectRequestedUnavailable (if encountered while processing wsen:Enumerate) wsman:CannotProcessFilter (if encountered while processing wse:Subscribe)
WIPG0222	Query language feature not supported by WBEM service infrastructure	wsen:CannotProcessFilter (if encountered while processing wsen:Enumerate) wsman:CannotProcessFilter (for exceptions encountered while processing wse:Subscribe)
WIPG0223	Invalid query	wsen:CannotProcessFilter (if encountered while processing wsen:Enumerate) wsman:CannotProcessFilter (if encountered while processing wsen:Enumerate)
WIPG0227	Operation failure	wsman:InternalError
WIPG0228	Operation not supported by CIM class implementation	wsa:ActionNotSupported
WIPG0229	CIM method invocation not supported by WBEM service infrastructure	wsa:ActionNotSupported

## 2486 19 XSD

2487 A normative copy of the XML schemas ([XML Schema Part 1](#), [XML Schema Part 2](#)) for this specification  
2488 may be retrieved by resolving the XML namespace URIs for this specification (listed in clause 5).

## 2489 20 WSDL

2490 This specification does not define a normative WSDL document. While it is possible to define a generic  
2491 WSDL document that can apply to all CIM classes, it does a disservice to developers who can provide a  
2492 more specific WSDL document tailored to a specific CIM class.

2493 **R20-1:** WSDL documents for a CIM class should include all WS-Management 1.1 resource access  
2494 operations.

2495 **R20-2:** WSDL documents for a CIM class or the query engine should include all WS-Management 1.1  
2496 Enumeration operations.

2497 **R20-3:** WSDL documents for a CIM class or the query engine should include all WS-Management 1.1  
2498 notifications operations.

2499 **R20-4:** WSDL documents for a CIM class should include operations for all extrinsic methods defined

2500 by the class.

2501

**ANNEX A  
(informative)****Change Log**

Version	Date	Description
1.0.0	2009-06-19	Released as DMTF Standard
1.1.0	2010-03-03	Released as DMTF Standard, with the following changes: <ul style="list-style-type: none"><li>• Addressed consistency issues with DSP0226</li></ul>
1.2.0	2011-06-30	Released as DMTF Standard, with the following changes: <ul style="list-style-type: none"><li>• Included ability to return URLs instead of full EPRs for metadata</li><li>• Included examples of metadata retrievals</li></ul>

2506

2507

2508

## Bibliography

2509 DMTF DSP8016, *WBEM Operations Message Registry, 1.0 Preliminary*,  
2510 <http://schemas.dmtf.org/wbem/messageregistry/1/dsp8016.xml>

2511

2512

2513